TITOC Working Group
Internet Draft
Intended status: Standards
Expires: March 22, 2011

S. Davari
A. Oren
Broadcom
L. Martini
Cisco

Sep 22, 2010

Transporting PTP messages (1588) over MPLS Networks
draft-davari-tictoc-1588overmpls-00.txt

Status of this Memo

Copyright Notice

Abstract

This document defines the method for transporting PTP messages (PDUs) over an MPLS network to enable a proper handling of these packets (e.g. implementation of Transparent Clocks (TC)) in LSRs.

The basic idea is to transport PTP messages inside dedicated MPLS LSPs. These LSPs only carry PTP messages and possibly Control and Management packets, but they do not carry customer traffic.

Two methods for transporting 1588 over MPLS are defined. The first method is to transport PTP messages directly over the dedicated MPLS LSP via UDP/IP encapsulation, which is suitable for IP/MPLS networks. The second method is to transport PTP messages inside a PW via Ethernet encapsulation, which is more suitable for MPLS-TP networks.

Table of Contents

1. Introduction

The objective of Precision Time Protocol (PTP) is to synchronize independent clocks running on separate nodes of a distributed system. [IEEE1588] defines PTP messages for clock and time synchronization. The PTP messages include PTP PDUs over UDP/IP (Annex D & E of [IEEE1588]) and PTP PDUs over Ethernet (Annex F of [IEEE1588]). This

document defines mapping and transport of the PTP messages defined in [IEEE1588] over MPLS networks.

PTP defines intermediate clock functions (called transparent clocks) between the source of time (Master) and the Slave clocks. Boundary Clocks (BC) form Master-Slave hierarchy with the Master clock as root. The messages related to synchronization, establishing the Master-Slave hierarchy, and signaling, terminate in the protocol engine of a boundary clock and are not forwarded. Management messages however, are forwarded to other ports on the boundary clock.

Transparent clocks modify a "correction field" (CF) within the synchronization messages to compensate for residence and propagation delays. Transparent clocks do not terminate synchronization, Master-Slave hierarchy control messages or signaling messages.

There is a need to transport PTP messages over MPLS networks. The MPLS network could be a transit network between 1588 Masters and Slaves. The accuracy of the recovered clock improves and the Slave logic simplifies when intermediate nodes (e.g. LSRs) properly handle PTP messages (e.g. perform TC), otherise the jitter at the 1588 Slave may be excessive and therefore the Slave may not be able to properly rcover the clock and time of day.

This document requires that MPLS nodes (LSRs) SHOULD be able to support the Transparent Clock (TC) function, meaning that they should be able to modify the CF of the proper PTP messages, via a 1-step or 2-step process. Such LSR is called "1588-aware LSR" in this document.

TC requires a 1588-aware LSR in the middle of an LSP to identify the PTP messages and perform proper update of the CF.

More generally this document requires that an LSR SHOULD be able to properly handle the PTP messages. For instance for those cases when the TC function is not viable (e.g. due to layer violation) as an alternative it should be possible to instead control the delay for these messages on both directions across the node.

In the above cases it is beneficial that PTP packets can be easily identified when carried over MPLS.

This document provides two methods for transporting PTP messages over MPLS. The main objectives are for LSRs to be able to deterministically detect and identify the PTP messages.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

3. Terminology

1588: The timing and synchronization as defined by IEEE 1588

PTP: The timing and synchronization protocol used by 1588

Master: The Source of 1588 Timing and clock

Slave: The Destination of 1588 Timing and clock that tries to follow the Master clock.

OC: Ordinary Clock

TC: Transparent Clocking, a time stamping method applied by intermediate nodes between Master and Slave

BC: Border Clock, is a node that recovers the Master clock via a Slave function and uses that clock as the Master for other Slaves.

PTP LSP: An LSP dedicated to carry PTP messages

PTP PW: A PW within a PTP LSP that MAY correspond to a Master/Slave flow.

CW: Pseudo wire Control Word

HW: Hardware

LAG: Link Aggregation

ECMP: Equal Cost Multipath

CF: Correction Field, a field inside certain PTP messages (message type 0-3)that holds the accumulative transit time inside intermediate switches

4. Problem Statement

When PTP messages are transported over MPLS networks, there is a need for intermediate LSRs to detect such messages and perform proper processing (e.g. Transparent Clock (TC)). Note the TC processing could be in the form of 1-Step or 2-Step time stamping.

PTP messages over Ethernet or IP can always be tunneled over MPLS. However the 1588 over MPLS mapping defined in this document is applicable whenever MPLS LSRs are 1588-aware and the intention is for those LSRs to perform proper processing on these packets.

When 1588-awareness is needed PTP messages should NOT be transported over LSPs or PWs that are carrying customer traffic because LSRs perform Label switching based on the top label in the stack. To detect PTP messages inside such LSPs require special Hardware (HW) to do deep packet inspection at line rate. Even if one assumes a deep packet inspection HW at line rate exists, the payload can’t be deterministically identified by LSRs because the payload type is a context of the PW label and the PW label and its context are only known to the Edge routers (PEs) and LSRs don’t know what is a PW’s payload (Ethernet, ATM, FR, CES, etc). Even if one assumes only Ethernet PWs are permitted in an LSP, the LSRs don’t have the knowledge of whether PW Control Word (CW) is present or not and therefore can’t deterministically identify the payload.

Therefore a generic method is defined in this document that does not require deep packet inspection at line rate, and can deterministically identify PTP messages. The defined method is applicable to both MPLS and MPLS-TP networks.

5. Dedicated LSPs for PTP messages

The method defined in this document can be used by LSRs to identify PTP messages in MPLS tunnels by using dedicated LSPs to carry PTP messages.

Compliant implementations MUST use dedicated LSPs to carry PTP messages over MPLS. Let’s call these LSPs as the "PTP LSPs" and the labels associated with these LSPs as "PTP labels". These LSPs could be P2P or P2MP LSPs. The PTP LSP between Master and Slaves MAY be P2MP or P2P LSP while the PTP LSP between each Slave and Master SHOULD be P2P LSP. The PTP LSP between a Master and a Slave and the PTP LSP between the same Slave and Master MUST be co-routed. Alternatively, a single bidirectional co-routed LSP can be used. The PTP LSP MAY be MPLS LSP or MPLS-TP LSP.

The PTP LSPs could be configured or signaled via RSVP-TE/GMPLS. New RSVP-TE/GMPLS TLVs and objects are defined in this document to indicate that these LSPs are PTP LSPs.

Note that the PTP LSPs MUST only carry PTP messages and MAY carry MPLS/MPLS-TP control and management messages such as BFD and LSP-Ping.

6. 1588 over MPLS Encapsulation

This document defines two methods for carrying PTP messages over MPLS. The first method is carrying PTP messages over PTP LSPs and the second method is to carry PTP messages over dedicated Ethernet PWs (called PTP PWs) inside PTP LSPs.

6.1. 1588 over LSP Encapsulation

The simplest method of transporting PTP messages over MPLS is to encapsulate PTP PDUs in UDP/IP and then encapsulate them in PTP LSP. The 1588 over LSP format is shown in Figure 1.

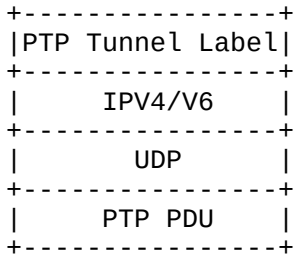

Figure 1 - 1588 over LSP Encapsulation

This encapsulation is very simple and is useful when the networks between 1588 Master and Slave are IP/MPLS networks.

In order for an LSR to process PTP messages, the PTP Label MUST be the top label of the label stack.

The UDP/IP encapsulation of PTP MUST follow Annex D and E of [IEEE1588].

6.2. 1588 over PW Encapsulation

Another method of transporting 1588 over MPLS networks is by encapsulating PTP PDUs in Ethernet and then transporting them over Ethernet PW (PTP PW) as defined in [RFC4448] , which in turn is transported over PTP LSPs. Alternatively PTP PDUs MAY be encapsulated in UDP/IP/Ethernet and then transported over Ethernet PW.

Both Raw and Tagged modes for Ethernet PW are permitted. The 1588 over PW format is shown in Figure 2.

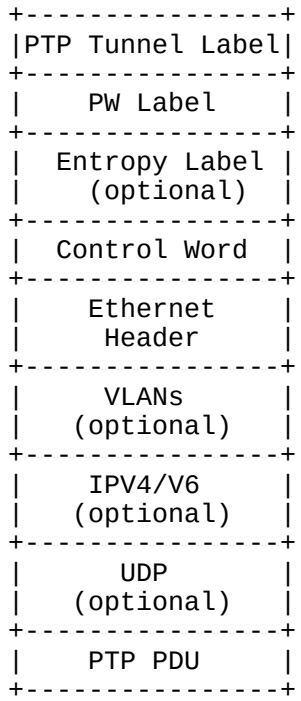

Figure 2 - 1588 over PW Encapsulation

The Control Word (CW) as specified in [RFC4448] is SHOULD be used to ensure a more robust detection of PTP messages inside the MPLS packet. If CW is used, the use of Sequence number is optional.

The use of VLAN and UDP/IP are optional. Note that 1 or 2 VLANs MAY exist in the PW payload.

In order for an LSR to process PTP messages, the top label of the label stack (the Tunnel Label) MUST be from PTP label range. However in some applications the PW label may be the top label in the stack, such as cases where there is only one-hop between PEs. In such cases, the PW label SHOULD be chosen from the PTP Label range.

An Entropy label [Fat PW] MAY be present at the bottom of stack.

The Ethernet encapsulation of PTP MUST follow Annex F of [IEEE1588] and the UDP/IP encapsulation of PTP MUST follow Annex D and E of [1EEE1588].

7. 1588 Message Transport

1588 protocol comprises of a number of message types. A subset of PTP messages that require TC processing are:

SYNC

FOLLOW_UP

DELAY_REQ (Delay Request)

DELAY_RES (Delay Response)

PDELAY_REQ (Peer Delay Request)

PDELAY_RESP (Peer Delay Response)

PDELAY_RESP_FOLLOW_UP (Peer Delay Response Follow up)

SYNC, FOLLOW_UP, DELAY_REQ and DELAY_RESP are exchanged between Master and Slave and MUST be transported over PTP LSPs.

PDELAY_REQ, PDELAY_RESP, and PDELAY_RESP_FOLLOW_UP are exchanged between adjacent routers and MAY be transported over PTP LSPs.

For a given instance of 1588 protocol SYNC, FOLLOW_UP, and DELAY_RESP MUST be transported over the same PTP LSP in the direction from Master to Slave, while DELAY_REQ MUST be transported over another PTP LSP in the reverse direction meaning in the direction from Slave to Master. These PTP LSPs, which are in opposite directions MUST be congruent and co-routed. Alternatively, a single bidirectional co-routed LSP can be used.

Other PTP message types are end-to-end messages between Master and Slave that don't need to be processed by intermediate routers. These message types MAY be carried in PTP Tunnel LSPs or any other LSP. When these PTP messages are carried in PTP LSPs there is no need to distinguish between the PTP message types, since the CF of these messages will be ignored by Slave clock.

8. Protection and Redundancy

In order to ensure continuous uninterrupted operation of 1588 Slaves, usually as a general practice, Redundant Masters are tracked by each Slave. It is the responsibility of the network operator to ensure that physically disjoint PTP tunnels that don't share any link are used between the redundant Masters and a Slave.

When redundant Masters are tracked by a Slave, any PTP LSP or PTP PW failure will trigger the slave to switch to the Redundant Master. However LSP/PW protection such as Linear Protection Switching (1:1, 1+1), Ring protection switching or MPLS Fast Reroute (FRR) SHOULD still be used to ensure the LSP/PW is ready for a future failure.

Note that any protection or reroute mechanism that adds additional label to the label stack, such as Facility Backup Fast Reroute, MUST ensure that the pushed label is a PTP Label to ensure proper processing of PTP messages by LSRs in the backup path.

9. ECMP and LAG

To ensure the proper operation of 1588 Slaves, the physical path for PTP messages from Master to Slave and vice versa MUST be the same for all PTP messages listed in section 7 and MUST not change even in presence of ECMP and LAG in the MPLS network.

The network operator MUST either ensure that the ECMP or LAG hashing algorithms keep the PTP messages described in section 7 and belonging to the same 1588 flow on the same link and path, or MUST disable LAG and/or ECMP for the PTP LSPs and/or PWs.

10. OAM, Control and Management

In order to manage PTP LSPs and PTP PWs, they MAY carry OAM, Control and Management messages. These control and management messages can be differentiated from PTP messages via already defined IETF methods.

In particular BFD [RFC5880], [RFC5884] and LSP-Ping [RFC4389] MAY run over PTP LSPs via UDP/IP encapsulation or via GAL/G-ACH. These Management protocols are easily identified by the UDP Destination Port number or by GAL/ACH respectively.

Also BFD, LSP-Ping and other Management messages MAY run over PTP PW via one of the defined VCCVs (Type 1, 2 or 3) [RFC5085]. In this case G-ACH, Router Alert Label (RAL), or PW label (TTL=1) are used to identify such Management messages.

11. FCS Recalculation

Ethernet FCS MUST be recalculated at every LSR that performs the TC processing and FCS retention described in [RFC4720] MUST not be used.

12. RSVP-TE/GMPLS Extensions for support of 1588

RSVP-TE/GMPLS signaling MAY be used to setup the PTP LSPs. A new object or TLV is required to signal that this is a PTP LSP. The OFFSET from bottom of label stack to the start of the PTP PDU MAY also be signaled. The LSRs that receive and process the RSVP-TE/GMPLS messages MAY use the OFFSET to locate the PTP 'correction field' (CF).

Note that the new object/TLV Must be ignored by LSRs that are not compliant to this specification.

The signaling details will be added in future versions of the draft.

13. Backward compatibility with non-1588-aware LSRs

It is most beneficial that all LSRs in the path of a PTP LSP be 1588-aware LSRs. This would ensure teh highest quality time and clock synchronization by 1588 Slaves. However, this specification does not mandate that all LSRs in path of a PTP LSP be 1588-aware.

Non-1588-aware LSRs just switch the MPLS packets carrying 1588 messages as data packets.

14. Other considerations

The use of Explicit Null (Label= 0 or 2) is accepatble as long as either the Explicit Null label is the bottom of stack label (applicable only to UDP/IP encapsulation) or the label below the Explicit Null label is a PTP label.

The use of Penultimate Hop Poping (PHP) is acceptable as long as either the PHP label is the bottom of stack label (applicable only to UDP/IP encapsulation) or the label below the PHP label is a PTP label.

15. Security Considerations

MPLS PW security considerations in general are discussed in [RFC3985] and [RFC4447], and those considerations also apply to this document.

An experimental security protocol is defined in [1]. The PTP security extension and protocol provide group source authentication, message integrity, and replay attack protection for PTP messages.

16. IANA Considerations

A new TLV is required to signal that PTP LSPs. IANA needs to assign the new TLV Type.

17. References

17.1. Normative References

[IEEE1588] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, IEEE 1588-2008

[RFC4448] Martini, L., Rosen, E., El-Aawar, and G.Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", April 2006.

[RFC4389] K. Kompella, G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", February 2006

[RFC5085] T. Nadeau, C. Pignataro "Pseudowire Virtual Circuit Connectivity Verification (VCCV):A Control Channel for Pseudowires", December 2007

[RFC5880] D. Katz, D. Ward, "Bidirectional Forwarding Detection", June 2010

[RFC3985] Bryant, S. and P. Pate, "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", March 2005

[RFC4447] Martini, L., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudo wire Setup and Maintenance Using the Label Distribution Protocol (LDP)", April 2006.

[RFC5884] R. Aggarwal, K. Kompella, T. Nadeau, G. Swallow, "Bidirectional Forwarding Detection for MPLS", June 2010

[RFC4720] A. Malis, D.Allan,N. Del Regno, "Pseudowire Emulation Edge-to-Edge (PWE3)Frame Check Sequence Retention", November 2006

17.2. Informative References

[Fat PW] S. Bryant, "Flow Aware Transport of Pseudowires over an MPLS PSN", January 2010

18. Acknowledgments

Authors' Addresses

Shahram Davari
Broadcom Corp.
3151 Zanker Road
San Jose, CA 95134

Email: davari@ieee.org
davari@boadcom.com

Amit Oren
Broadcom Corp.
3151 Zanker Road
San Jose, CA 95134

Email: amito@broadcom.com

Luca Martini
Cisco Systems
San Jose,CA

Email: Lmartini@cisco.com

TICTOC
Internet Draft
Intended status: Informational
Expires: January 5, 2011

Tim Frost,
Greg Dowd,
Symmetricom, Inc.

Laurent Montini,
Cisco Systems

July 5, 2010

Management Requirements for Packet-based Timing Distribution
draft-frost-tictoc-management-00.txt

Status of this Memo

Copyright Notice

Abstract

This Internet draft investigates the management aspects associated with packet-based distribution of time and frequency using protocols such as PTP (Precision Time Protocol, [1]). It explores some of the issues that need to be solved in connection with the management of synchronization distribution.

Table of Contents

1. Introduction

Synchronization for many telecoms applications (e.g. wireless basestations, circuit emulation services) is a mission-critical service, in the sense that if the synchronization service goes out of tolerance, the enabled service may fail, impacting revenue. When the synchronization is delivered by a packet-based mechanism (e.g. by use of PTP defined in [1]), continuous in-service monitoring is required to verify the quality and traceability of the synchronization.

The purpose of this draft is to examine some of the requirements of synchronization management and to propose options for how these issues may be tackled. It has been developed out of the informal "Problem Statement for Management of Synchronization Networks" presented at IETF 77.

1.1. Elements of synchronization management

Elements in effective management and monitoring for packet-based synchronization distribution include:

- Fault monitoring and reporting
- Performance and status monitoring of the synchronization equipment
- Performance and status monitoring of the packet network related to timing distribution

Analysis of the performance data for trends in key synchronization performance indicators may allow "early warning" of possible issues (e.g. congestion) that may affect synchronization. Continuous, in-service monitoring enables the operator to be informed of events or trends likely to affect the synchronization network and enable corrective action to be taken.

1.2. Use of a single synchronization management domain

Whilst distributed across the network, and possibly embedded into disparate network elements, synchronization forms a distinct infrastructural function within the network. This means it needs to be planned and managed as an entity, and not as collection of separate components.

The aggregation of synchronization information and processing of it as an integrated whole can provide powerful insights into the overall performance of the synchronization service, and indicate if more general corrective action is required. For example, degradation in the key performance indicators of several synchronization network elements may be an early warning sign of increased network loading.

Use of specific synchronization node manager can enhance such holistic management of the synchronization function. It also simplifies the integration of the synchronization management into an operator’s OSS (Operations and Support System), by providing a single point of integration with visibility of the whole network, including the synchronization service, and allowing correlation of information from multiple network information.

2. Issues to be resolved

Some of the issues that need to be resolved in the creation of a coherent approach to synchronization management include:

- What information must be maintained by synchronization functions?
- What performance data related to synchronization are to be collected?
- What alarms must be generated by synchronization functions?
- How is the management information to be collected?
- How can network elements containing synchronization functions be discovered?

These issues are discussed in the following sections.

2.1. What information must be maintained by synchronization functions?

A standard set of "synchronization information" should be defined, such that all synchronization functions are able to report the same types of information. This should include node information related to timing and synchronization, protocol-specific information (e.g. for PTP-based functions, the standard data sets) and timing performance data, enabling a synchronization network manager to assess the health of a synchronization node.

The standard set of information should be defined in terms of a MIB (Management Information Base) for each type of synchronization function (e.g. packet master or slave clock, or "on-path" timing support elements).

2.2. What performance data related to the timing flow are to be collected?

A standard set of information relating to the quality and performance of the timing packet flow will enable a synchronization network manager to assess the health of a individual timing path and of the synchronization network as a whole.

The standard set of information could be defined in terms of a IPFIX Information Model using IPFIX protocol for collecting the information from various nodes.

2.3. What alarms must be generated by synchronization functions?

Similarly, a standard set of alarms for synchronization functions should be defined. These should include conventional alarm criteria such as input signal failure, as well as more specific packet-based synchronization criteria, such as the PTSF conditions defined in the ITU-T’s Telecom Profile [2].

2.4. How is the management data to be collected?

Another consideration is how the data are to be collected. This may be dependent on the equipment in which the synchronization functions are embedded, the type of information, and the operator's own management strategy. Some potential options include:

- through a management channel in the synchronization flow (e.g. PTP management messages), to a synchronization network manager
- through a management channel distinct from the synchronization flow (e.g. SNMP or IPFIX protocols)
- through the element management system of a network element containing a synchronization function, and then northbound into the OSS
- through the element management system of a network element containing a synchronization function, and then northbound into a synchronization network manager

2.5. Identification of network elements containing synchronization functions

One of the main issues is to identify network elements containing synchronization functions. A synchronization network management system can only manage devices that it knows exist, and in a large network, it may be difficult to discover which network elements contain synchronization functions.

This identification process is not strictly speaking a management function, but it is relevant and necessary to enable on-going synchronization management. Some options for identification of synchronization functions include:

- synchronization function identifies itself to a pre-configured synchronization management node on startup
- synchronization masters or servers maintain a list of their currently serviced slaves/clients, and make the list available for the synchronization network manager to query.

3. Security Considerations

Security aspects of the above options will need to be considered in more detail.

4. IANA Considerations

No IANA actions are required as a result of the publication of this document.

5. Acknowledgements

The authors wish to thank Sanjay Mani (Symmetricom) for his invaluable comments.

This document was prepared using 2-Word-v2.0.template.dot.

6. Informative References

[1] IEEE, "Standard for A Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE1588-2008.

[2] "ITU-T PTP Profile for Frequency distribution without timing support from the network ", Draft Recommendation G.8265.1 (work in progress), TD-PLEN-0255-R1, June 2010

Author's Addresses

Tim Frost,
Symmetricom Inc.,
2300 Orchard Parkway,
San Jose,
CA 95131,
USA.
Email: tfrost@symmetricom.com

Greg Dowd,
Symmetricom Inc.,
2300 Orchard Parkway,
San Jose,
CA 95131,
USA.
Email: gdowd@symmetricom.com

Laurent Montini,
Cisco Systems,
11, rue Camille Desmoulins,
92782 Issy-les-Moulineaux,
France.
Email: lmontini@cisco.com

TICTOC
Internet Draft
Intended status: Informational
Expires: January 5, 2011

Tim Frost
Greg Dowd
Symmetricom

July 5, 2010

Definitions of Managed Objects for Precision Time Protocol Version 2 (PTPv2) Slave Clocks

draft-frost-tictoc-ptp-slave-mib-00.txt

Status of this Memo

This Internet-Draft will expire on January 5, 2011.

Copyright Notice

Abstract

This draft contains a preliminary MIB definition for a PTP Slave-Only Ordinary clock. This targeted at a slave clock compliant with the PTP Telecom Profile described in [G.8265.1].

Table of Contents

1. Introduction

A companion draft to this [Fro2010] addresses the issues in managing packet-based synchronization systems, and highlights some areas that need further work and development. This draft addresses the points raised in clauses 2.1 and 2.2 of [Fro2010], by defining in MIB form the synchronization information and performance data that a PTP slave device is expected to maintain.

The PTPv2 Slave MIB module is targeted at the Slave-Only Ordinary Clock instance described in the ITU's PTPv2 Telecom Profile for Frequency Distribution [G.8265.1]. It contains the standard data sets defined in [IEEE1588-2008], plus some additional data relating to status and performance of the clock servo loop.

The MIB contained in this version of the draft is not yet complete, nor is it totally syntactically correct. However, it is an initial version sufficient to show the intent and direction of development.

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410]. Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the

SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

3. Technical Description

The following standard data sets must be maintained in a PTPv2 Slave-Only Ordinary Clock. Each of these data sets is described in clause 8.2 of [IEEE1588-2008].

- Default Data Set
- Current Data Set
- Parent Data Set
- Time Properties Data Set
- Port Data Set

These data sets provide information on the protocol aspects of a clock, but do not provide information on the status or performance of the clock servo loop. For a slave-only ordinary clock, this includes the following additional data sets:

- Slave Status Data Set (contains current status information on the clock servo loop)
- Slave Performance Data Set (contains performance information on the clock servo loop)
- Alarm Status Data Set (contains the alarm status of the device)
- Network Statistics Data Set (contains performance information on the network itself)

Each of these data sets is represented within the MIB using standard MIB syntax.

4. MIB Definition

```
   -- ****************************************************************
   --
   --    The PTPv2 Slave-Only Ordinary Clock
   --    Management Information Base (MIB)
   --
   --    For Slave-Only Ordinary Clocks compliant with the
   --    ITU PTPv2 Telecom Profile for Frequency Distribution
   --    [G.8265.1], Annex A
   --
   --    Authors: Tim Frost (tfrost@symmetricom.com)
   --             Greg Dowd (gdowd@symmetricom.com)
   --
   --    for the Internet Engineering Task Force (IETF)
   --    TICTOC Working Group (TICTOC)
   --
   -- ****************************************************************
   --
   --    Rev 0.0
   --    Published as draft-frost-tictoc-ptp-slave-mib-00.txt
   --
   -- ****************************************************************

   PTPV2-SLAVE-MIB DEFINITIONS ::= BEGIN

   IMPORTS
      enterprises,
      mib-2,
      MODULE-IDENTITY,
      OBJECT-TYPE,
      IpAddress
         FROM SNMPv2-SMI
      MacAddress,
      TEXTUAL-CONVENTION
         FROM SNMPv2-TC
      OBJECT-GROUP,
      NOTIFICATION-GROUP
         FROM SNMPv2-CONF;

   ptpRegMIB MODULE-IDENTITY
      LAST-UPDATED "201006301146Z"
      ORGANIZATION "The IETF TICTOC Working Group (tictoc)"
      CONTACT-INFO
         "WG Email: tictoc@ietf.org

         Tim Frost,
         Symmetricom Inc.,
         Email: tfrost@symmetricom.com
```

```
         Greg Dowd,
         Symmetricom Inc.,
         Email: gdowd@symmetricom.com"
      DESCRIPTION
         "The Management Information Base for
         PTPv2 Slave-Only Ordinary Clocks

         Copyright (c) 2010 IETF Trust and the persons identified
         as the document authors.  All rights reserved.

         Redistribution and use in source and binary forms, with
         or without modification, is permitted pursuant to, and
         subject to the license terms contained in, the
         Simplified BSD License set forth in Section 4.c of the
         IETF Trusts Legal Provisions relating to IETF Documents
         (http://trustee.ietf.org/license-info)."
      REVISION "201006301146Z"
      DESCRIPTION
         "This revision of the MIB is published as
         draft-frost-tictoc-ptp-slave-mib-00.txt"
      ::= { mib-2 1 }


   PtpClockIdentity ::= TEXTUAL-CONVENTION
      STATUS  current
      DESCRIPTION
         "The clockIdentity is an 8-octet array.
         The value of the clockIdentity should be taken from the
         IEEE EUI-64 individual assigned numbers.
         (3 octet OUI, followed by a 5 octet extension)"
      SYNTAX OCTET STRING


   -- Scalars and Tables
   --

   ptpObjects OBJECT IDENTIFIER  ::= { ptpRegMIB 1 }

   --
   --Clock description from clause 15.5.3.1.2 of IEEE1588-2008
   --
   ptpClockDescription OBJECT IDENTIFIER  ::= { ptpObjects 1 }

   ptpClockType OBJECT-TYPE
      SYNTAX  BITS {
            ordinary(0),
            boundary(1),
            transparentP2P(2),
```

```
            transparentE2E(3),
            management(4) }
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of clockType shall indicate the type of PTP node
         as defined in Table 42 of IEEE1588-2008."
      REFERENCE
         "Clause 15.5.3.1.2.1 of IEEE1588-2008"
      DEFVAL { { ordinary } }
      ::= { ptpClockDescription 1 }


   ptpPhysicalLayerProtocol OBJECT-TYPE
      SYNTAX  OCTET STRING
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of physicalLayerProtocol shall indicate the
         physical layer protocol defining the physicalAddress member."
      REFERENCE
         "Clause 15.5.3.1.2.2 of IEEE1588-2008"
      ::= { ptpClockDescription 2 }


   ptpPhysicalAddressLength OBJECT-TYPE
      SYNTAX  INTEGER (1..16)
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of physicalAddressLength is the number of octets
         in the physicalAddress field.
         The range shall be 1 to 16 octets."
      REFERENCE
         "Clause 15.5.3.1.2.3 of IEEE1588-2008"
      ::= { ptpClockDescription 3 }


   ptpPhysicalAddress OBJECT-TYPE
      SYNTAX  MacAddress
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of physicalAddress shall be the physical address
         of the port indicated by the targetPortIdentity.portNumber
         member of the field, for example, the MAC address for an
         IEEE 802.3 end station."
      REFERENCE
         "Clause 15.5.3.1.2.4 of IEEE1588-2008"
```

```
      ::= { ptpClockDescription 4 }


   ptpProtocolAddress OBJECT-TYPE
      SYNTAX  IpAddress
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of protocolAddress shall be the protocol address
         of the port indicated by the targetPortIdentity.portNumber
         member of the field."
      REFERENCE
         "Clause 15.5.3.1.2.5 of IEEE1588-2008"
      ::= { ptpClockDescription 5 }


   ptpManufacturerIdentity OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of manufacturerIdentity shall be an OUI
         owned by the manufacturer of the node."
      REFERENCE
         "Clause 15.5.3.1.2.6 of IEEE1588-2008"
      ::= { ptpClockDescription 6 }


   ptpProducDescription OBJECT-TYPE
      SYNTAX  OCTET STRING
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The productDescription field shall indicate, in order:
         a) The name of the manufacturer of the node,
            manufacturerName, followed by a semicolon (;)
         b) The model number of the node,
            modelNumber, followed by a semicolon(;)
         c) An identifier of the instance of this mode,
            instanceIdentifier, such as the MAC address
            or the serial number
         The maximum number of symbols in the
         productDescription text field shall be 64."
      REFERENCE
         "Clause 15.5.3.1.2.7 of IEEE1588-2008"
      ::= { ptpClockDescription 7 }


   ptpRevisionData OBJECT-TYPE
```

```
      SYNTAX  OCTET STRING
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value shall indicate the revisions for node
         hardware (HW), firmware (FW), and software (SW).
         This information shall be semicolon (;) separated
         text fields in the order HW;FW;SW.
         Nonapplicable elements shall be indicated by a text
         field of zero length."
      REFERENCE
         "Clause 15.5.3.1.2.8 of IEEE1588-2008"
      ::= { ptpClockDescription 8 }


   ptpUserDescription OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The userDescription field shall indicate, in order:
         a) A user-defined name of the device, e.g., Sensor-1,
            followed by a semicolon (;)
         b) A user-defined physical location of the device,
            e.g., Rack-2 Shelf-3
         Either field may be absent. By default no text is required."
      REFERENCE
         "Clause 15.5.3.1.2.9 of IEEE1588-2008"
      ::= { ptpClockDescription 9 }


   ptpProfileIdentity OBJECT-TYPE
      SYNTAX  OCTET STRING
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of profileIdentity shall identify the PTP profile
         implemented by the port indicated by the
         targetPortIdentity.portNumber member of the field."
      REFERENCE
         "Clause 15.5.3.1.2.10 of IEEE1588-2008"
      ::= { ptpClockDescription 10 }


   --
   --Default data set described in clause 8.2.1 of IEEE1588-2008
   --
   ptpDefaultDataSet OBJECT IDENTIFIER    ::= { ptpObjects 2 }
```

```
   ptpTwoStepFlag OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of defaultDS.twoStepFlag shall be TRUE
         if the clock is a two-step clock;
         otherwise, the value shall be FALSE."
      REFERENCE
         "Clause 8.2.1.2.1, IEEE1588-2008"
      ::= { ptpDefaultDataSet 1 }


   ptpClockIdentity OBJECT-TYPE
      SYNTAX  PtpClockIdentity
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of defaultDS.clockIdentity shall be the
         clockIdentity of the local clock."
      REFERENCE
         "Clause 8.2.1.2.2, IEEE1588-2008"
      ::= { ptpDefaultDataSet 2 }


   ptpNumberPorts OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "number of ports"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of defaultDS.numberPorts shall be the number of
         PTP ports on the device.
         For an ordinary clock, this value shall be 1."
      REFERENCE
         "Clause 8.2.1.2.3, IEEE1588-2008"
      DEFVAL { 1 }
      ::= { ptpDefaultDataSet 3 }


   ptpClockQuality OBJECT IDENTIFIER   ::= { ptpDefaultDataSet 5 }

   ptpClockClass OBJECT-TYPE
      SYNTAX  INTEGER {
            qlprs(80),     -- Primary Reference Source (Option II)
            qlstu(82),     -- Synchronization Traceability Unknown
            qlprc(84),     -- Primary Reference Clock (Option I)
            qlst2(86),     -- Stratum 2 (Option II)
```

```
            qlssua(90),    -- Primary level SSU (SSU-A, Option I),
   also Transit Node Clock (TNC, Option II)
            qlssub(96),    -- Second level SSU (SSU-B, Option I)
            qlst3e(100),   -- Stratum 3E (Option II)
            qlst3(102),    -- Stratum 3, also Ethernet Equipment Clock
   (Option II)
            qlsec(104),    -- SDH Equipment Clock, also Ethernet
   Equipment Clock (Option I)
            qlsmc(106),    -- SONET Minimum Clock (Option II)
            qlprov(108),   -- Provisionable by operator (Option II)
            qldnu(110)     -- Do Not Use (DNU, Option I), also Don't
   Use for Sync (DUS, Option II)
            }
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The clockClass attribute of an ordinary or boundary clock
         denotes the traceability of the time or frequency distributed
         by the grandmaster clock.
         The interpretation and allowed values of clockClass shall be
         based on the definitions in Clause 6.7.3.2 of G.8265.1
         (ITU PTPv2 Telecom Profile for Frequency Distribution)."
      REFERENCE
         "Clause 7.6.2.4, IEEE1588-2008
         Clause 6.7.3.2, G.8265.1"
      ::= { ptpClockQuality 1 }


   ptpClockAccuracy OBJECT-TYPE
      SYNTAX  INTEGER {
            ns25(32),      -- The time is accurate to 25ns
            ns100(33),     -- The time is accurate to 100ns
            ns250(34),     -- The time is accurate to 250ns
            us1(35),       -- The time is accurate to 1us
            us2p5(36),     -- The time is accurate to 2.5us
            us10(37),      -- The time is accurate to 10us
            us25(38),      -- The time is accurate to 25us
            us100(39),     -- The time is accurate to 100us
            us250(40),     -- The time is accurate to 250us
            ms1(41),       -- The time is accurate to 1ms
            ms2p5(42),     -- The time is accurate to 2.5ms
            ms10(43),      -- The time is accurate to 10ms
            ms25(44),      -- The time is accurate to 25ms
            ms100(45),     -- The time is accurate to 100ms
            ms250(46),     -- The time is accurate to 250ms
            s1(47),        -- The time is accurate to 1s
            s10(48),       -- The time is accurate to 10s
            s10plus(49)    -- The time is accurate to >10s
            }
```

```
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The clockAccuracy characterizes a clock for the
         purpose of the best master clock (BMC) algorithm.
         The value of clockAccuracy shall be taken from the
         enumeration in Table 6 of IEEE15888-2008."
      REFERENCE
         "Clause 7.6.2.5, IEEE1588-2008"
      ::= { ptpClockQuality 2 }


   ptpOffsetScaledLogVariance OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of offsetScaledLogVariance shall be computed and
         represented as described in 7.6.3 of IEEE1588-2008.
         The value is an estimate of the variations of the local
         clock from a linear timescale when it is not synchronized
         to another clock using the protocol."
      REFERENCE
         "Clause 7.6.3.5 of IEEE1588-2008"
      ::= { ptpClockQuality 3 }


   ptpPriority1 OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of defaultDS.priority1 is the priority1 attribute
         (see 7.6.2.2 of IEEE1588-2008) of the local clock."
      REFERENCE
         "Clause 8.2.1.4.1, IEEE1588-2008"
      ::= { ptpDefaultDataSet 6 }


   ptpPriority2 OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of defaultDS.priority2 is the priority2 attribute
         (see 7.6.2.3 of IEEE1588-2008) of the local clock."
      REFERENCE
         "Clause 8.2.1.4.2, IEEE1588-2008"
      ::= { ptpDefaultDataSet 7 }
```

```
   ptpDomainNumber OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of defaultDS.domainNumber is the domain attribute
         (see 7.1 of IEEE1588-2008) of the local clock."
      REFERENCE
         "Clause 8.2.1.4.3, IEEE1588-2008"
      ::= { ptpDefaultDataSet 8 }


   ptpSlaveOnly OBJECT-TYPE
      SYNTAX  TruthValue
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of defaultDS.slaveOnly shall be TRUE if the clock
         is a slave-only clock; see 9.2.2.
         The value shall be FALSE if the clock is a non-slave-only
   clock;
         see 9.2.3 of IEEE1588-2008."
      REFERENCE
         "Clause 8.2.1.4.4, IEEE1588-2008"
      DEFVAL { true }
      ::= { ptpDefaultDataSet 9 }


   --
   --Current data set described in clause 8.2.2 of IEEE1588-2008
   --
   ptpCurrentDataSet OBJECT IDENTIFIER    ::= { ptpObjects 3 }

   ptpStepsRemoved OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of currentDS.stepsRemoved is the number of
         communication paths traversed between the local clock
         and the grandmaster clock."
      REFERENCE
         "Clause 8.2.2.2, IEEE1588-2008"
      DEFVAL { 0 }
      ::= { ptpCurrentDataSet 1 }
```

```
   ptpOffsetFromMaster OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of currentDS.offsetFromMaster is an
         implementation-specific representation of the
         current value of the time difference between a
         master and a slave as computed by the slave."
      REFERENCE
         "Clause 8.2.2.3, IEEE1588-2008"
      ::= { ptpCurrentDataSet 2 }


   ptpMeanPathDelay OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of currentDS.meanPathDelay is an
         implementation-specific representation of the
         current value of the mean propagation time between
         a master and slave clock as computed by the slave."
      REFERENCE
         "Clause 8.2.2.4, IEEE1588-2008"
      ::= { ptpCurrentDataSet 3 }


   --
   --Parent data set described in clause 8.2.3 of IEEE1588-2008
   --
   ptpParentDataSet OBJECT IDENTIFIER  ::= { ptpObjects 4 }

   ptpParentPortIdentity OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of parentDS.parentPortIdentity is the portIdentity
         of the port on the master that issues the Sync messages used
         in synchronizing this clock."
      REFERENCE
         "Clause 8.2.3.2, IEEE1588-2008"
      ::= { ptpParentDataSet 1 }


   ptpParentStats OBJECT-TYPE
      SYNTAX  TruthValue
      MAX-ACCESS read-only
```

```
      STATUS  current
      DESCRIPTION
         "The value of parentDS.parentStats shall be TRUE
         if all of the following conditions are satisfied:
         - The clock has a port in the SLAVE state.
         - The clock has computed statistically valid estimates of the
           parentDS.observedParentOffsetScaledLog Variance and the
           parentDS.observedParentClockPhaseChangeRate members.
         - Otherwise the value shall be FALSE."
      REFERENCE
         "Clause 8.2.3.3, IEEE1588-2008"
      DEFVAL { false }
      ::= { ptpParentDataSet 2 }


   ptpObservedParentOffsetScaledLogVariance OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of parentDS.observedParentOffsetScaledLogVariance
         shall be an estimate of the parent clocks PTP variance as
         observed by the slave clock, computed and represented as
         described in 7.6.3.5 of IEEE1588-2008.."
      REFERENCE
         "Clause 8.2.3.4, IEEE1588-2008"
      ::= { ptpParentDataSet 3 }


   ptpObservedParentClockPhaseChangeRate OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of parentDS.observedParentClockPhaseChangeRate
         shall be an estimate of the parent clocks phase change rate
         as observed by the slave clock as defined in 7.6.4.4. of
         IEEE1588-2008.
         If the estimate exceeds the capacity of its data type,
         this value shall be set to 0x7FFF FFFF or 0x8000 0000,
         as appropriate."
      REFERENCE
         "Clause 8.2.2.5, IEEE1588-2008"
      ::= { ptpParentDataSet 4 }


   ptpGrandmasterIdentity OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
```

```
      STATUS  current
      DESCRIPTION
         "The value of parentDS.grandmasterIdentity is the
         clockIdentity attribute (see 7.6.2.1 of IEEE1588-2008)
         of the grandmaster clock."
      REFERENCE
         "Clause 8.2.3.6, IEEE1588-2008"
      ::= { ptpParentDataSet 5 }


   ptpGrandMasterClockQuality OBJECT IDENTIFIER    ::= {
   ptpParentDataSet 7 }

   ptpClockAccuracy0 OBJECT-TYPE
      SYNTAX  INTEGER {
            ns25(32),      -- The time is accurate to 25ns
            ns100(33),     -- The time is accurate to 100ns
            ns250(34),     -- The time is accurate to 250ns
            us1(35),       -- The time is accurate to 1us
            us2p5(36),     -- The time is accurate to 2.5us
            us10(37),      -- The time is accurate to 10us
            us25(38),      -- The time is accurate to 25us
            us100(39),     -- The time is accurate to 100us
            us250(40),     -- The time is accurate to 250us
            ms1(41),       -- The time is accurate to 1ms
            ms2p5(42),     -- The time is accurate to 2.5ms
            ms10(43),      -- The time is accurate to 10ms
            ms25(44),      -- The time is accurate to 25ms
            ms100(45),     -- The time is accurate to 100ms
            ms250(46),     -- The time is accurate to 250ms
            s1(47),        -- The time is accurate to 1s
            s10(48),       -- The time is accurate to 10s
            s10plus(49)    -- The time is accurate to >10s
            }
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The clockAccuracy characterizes a clock for the
         purpose of the best master clock (BMC) algorithm.
         The value of clockAccuracy shall be taken from the
         enumeration in Table 6 of IEEE15888-2008."
      REFERENCE
         "Clause 7.6.2.5, IEEE1588-2008"
      ::= { ptpGrandMasterClockQuality 2 }


   ptpClockClass0 OBJECT-TYPE
      SYNTAX  INTEGER {
            qlprs(80),     -- Primary Reference Source (Option II)
```

```
            qlstu(82),     -- Synchronization Traceability Unknown
            qlprc(84),     -- Primary Reference Clock (Option I)
            qlst2(86),     -- Stratum 2 (Option II)
            qlssua(90),    -- Primary level SSU (SSU-A, Option I),
   also Transit Node Clock (TNC, Option II)
            qlssub(96),    -- Second level SSU (SSU-B, Option I)
            qlst3e(100),   -- Stratum 3E (Option II)
            qlst3(102),    -- Stratum 3, also Ethernet Equipment Clock
   (Option II)
            qlsec(104),    -- SDH Equipment Clock, also Ethernet
   Equipment Clock (Option I)
            qlsmc(106),    -- SONET Minimum Clock (Option II)
            qlprov(108),   -- Provisionable by operator (Option II)
            qldnu(110)     -- Do Not Use (DNU, Option I), also Don't
   Use for Sync (DUS, Option II)
            }
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The clockClass attribute of an ordinary or boundary clock
         denotes the traceability of the time or frequency distributed
         by the grandmaster clock.
         The interpretation and allowed values of clockClass shall be
         based on the definitions in Clause 6.7.3.2 of G.8265.1
         (ITU PTPv2 Telecom Profile for Frequency Distribution)."
      REFERENCE
         "Clause 7.6.2.4, IEEE1588-2008
         Clause 6.7.3.2, G.8265.1"
      ::= { ptpGrandMasterClockQuality 1 }


   ptpOffsetScaledLogVariance0 OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of offsetScaledLogVariance shall be computed and
         represented as described in 7.6.3 of IEEE1588-2008.
         The value is an estimate of the variations of the local
         clock from a linear timescale when it is not synchronized
         to another clock using the protocol."
      REFERENCE
         "Clause 7.6.3.5 of IEEE1588-2008"
      ::= { ptpGrandMasterClockQuality 3 }


   ptpGrandmasterPriority1 OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
```

```
      STATUS  current
      DESCRIPTION
         "The value of parentDS.grandmasterPriority1 is the
         priority1 attribute (see 7.6.2.2 of IEEE1588-2008)
         of the grandmaster clock."
      REFERENCE
         "Clause 8.2.3.8, IEEE1588-2008"
      ::= { ptpParentDataSet 8 }


   ptpGrandmasterPriority2 OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of parentDS.grandmasterPriority2 is the
         priority2 attribute (see 7.6.2.2 of IEEE1588-2008)
         of the grandmaster clock."
      REFERENCE
         "Clause 8.2.3.9, IEEE1588-2008"
      ::= { ptpParentDataSet 9 }


   --
   --Time Properties data set described in clause 8.2.4 of IEEE1588-
   2008
   --
   ptpTimePropertiesDataSet OBJECT IDENTIFIER   ::= { ptpObjects 5 }

   ptpCurrentUTCOffset OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "seconds"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "In PTP systems whose epoch is the PTP epoch, the value of
         timePropertiesDS.currentUtcOffset is the offset between
         TAI and UTC; otherwise the value has no meaning.
         The value shall be in units of seconds."
      REFERENCE
         "Clause 8.2.4.2, IEEE1588-2008"
      ::= { ptpTimePropertiesDataSet 1 }


   ptpCurrentUTCOffsetValid OBJECT-TYPE
      SYNTAX  TruthValue
      MAX-ACCESS read-only
      STATUS  current
```

```
      DESCRIPTION
         "The value of timePropertiesDS.currentUtcOffsetValid
         is TRUE if the timePropertiesDS.currentUtcOffset
         is known to be correct."
      REFERENCE
         "Clause 8.2.4.3, IEEE1588-2008"
      ::= { ptpTimePropertiesDataSet 2 }


   ptpLeap59 OBJECT-TYPE
      SYNTAX  TruthValue
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "In PTP systems whose epoch is the PTP epoch, a TRUE value
         for timePropertiesDS.leap59 shall indicate that the last
         minute of the current UTC day contains 59 seconds."
      REFERENCE
         "Clause 8.2.4.4, IEEE1588-2008"
      ::= { ptpTimePropertiesDataSet 3 }


   ptpLeap61 OBJECT-TYPE
      SYNTAX  TruthValue
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "In PTP systems whose epoch is the PTP epoch, a TRUE value
         for timePropertiesDS.leap61 shall indicate that the last
         minute of the current UTC day contains 61 seconds."
      REFERENCE
         "Clause 8.2.4.5, IEEE1588-2008"
      ::= { ptpTimePropertiesDataSet 4 }


   ptpTimeTraceable OBJECT-TYPE
      SYNTAX  TruthValue
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of timePropertiesDS.timeTraceable is TRUE
         if the timescale and the value of
         timePropertiesDS.currentUtcOffset are traceable to a
         primary reference; otherwise, the value shall be FALSE."
      REFERENCE
         "Clause 8.2.4.6, IEEE1588-2008"
      ::= { ptpTimePropertiesDataSet 5 }
```

```
   ptpFrequencyTraceable OBJECT-TYPE
      SYNTAX  TruthValue
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of timePropertiesDS.frequencyTraceable is TRUE
         if the frequency determining the timescale is traceable
         to a primary reference; otherwise, the value shall be FALSE."
      REFERENCE
         "Clause 8.2.4.7, IEEE1588-2008"
      ::= { ptpTimePropertiesDataSet 6 }


   ptpPTPTimescale OBJECT-TYPE
      SYNTAX  TruthValue
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of timePropertiesDS.ptpTimescale is TRUE
         if the clock timescale of the grandmaster clock
         (see 7.2.1 of IEEE1588-2008) is PTP and FALSE otherwise."
      REFERENCE
         "Clause 8.2.4.8, IEEE1588-2008"
      ::= { ptpTimePropertiesDataSet 7 }


   ptpTimeSource OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of timePropertiesDS.timeSource is the source
         of time used by the grandmaster clock."
      REFERENCE
         "Clause 8.2.4.9, IEEE1588-2008"
      ::= { ptpTimePropertiesDataSet 8 }


   --
   --Port data set described in clause 8.2.5 of IEEE1588-2008
   --
   ptpPortDataSet OBJECT IDENTIFIER    ::= { ptpObjects 6 }

   ptpPortIdentity OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of portDS.portIdentity shall be the PortIdentity
```

```
         attribute of the local port; see 7.5.2 of IEEE1588-2008."
      REFERENCE
         "Clause 8.2.5.2.1, IEEE1588-2008"
      ::= { ptpPortDataSet 1 }


   ptpPortState OBJECT-TYPE
      SYNTAX  INTEGER {
            initializing(1),
            faulty(2),
            disabled(3),
            listening(4),
            preMaster(5),
            master(6),
            passive(7),
            uncalibrated(8),
            slave(9) }
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of portDS.portState shall be the value of the
         current state of the protocol engine associated with this port
         (see 9.2 of IEEE1588-2008) and shall be taken from the
         enumeration in Table 8 of IEEE1588-2008."
      REFERENCE
         "Clause 8.2.3.5.2, IEEE1588-2008"
      DEFVAL { initializing }
      ::= { ptpPortDataSet 2 }


   ptpLogMinDelayReqInterval OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of portDS.logMinDelayReqInterval is the
         logarithm to the base 2 of the minDelayReqInterval;
         see 7.7.2.4 of IEEE1588-2008.
         The initialization value is implementation-specific
         consistent with 7.7.2.4."
      REFERENCE
         "Clause 8.2.5.3.2, IEEE1588-2008"
      ::= { ptpPortDataSet 3 }


   ptpPeerMeanPathDelay OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
```

```
      DESCRIPTION
         "If the value of the portDS.delayMechanism member is
         peer-to-peer (P2P), the value of portDS.peerMeanPathDelay
         shall be an estimate of the current one-way propagation
         delay on the link."
      REFERENCE
         "Clause 8.2.5.3.3, IEEE1588-2008"
      ::= { ptpPortDataSet 4 }


   ptpLogAnnounceInterval OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of portDS.logAnnounceInterval shall
         be the logarithm to the base 2 of the mean
         announceInterval; see 7.7.2.2 of IEEE1588-2008."
      REFERENCE
         "Clause 8.2.5.4.1, IEEE1588-2008"
      ::= { ptpPortDataSet 5 }


   ptpAnnounceReceiptTimeout OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of portDS.announceReceiptTimeout shall be an
         integral multiple of announceInterval;
         see 7.7.3.1 of IEEE1588-2008."
      REFERENCE
         "Clause 8.2.5.4.2, IEEE1588-2008"
      ::= { ptpPortDataSet 6 }


   ptpLogSyncInterval OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of portDS.logSyncInterval shall be the
         logarithm to the base 2 of the mean SyncInterval
         for multicast messages; see 7.7.2.3 of IEEE1588-2008."
      REFERENCE
         "Clause 8.2.5.4.3, IEEE1588-2008"
      ::= { ptpPortDataSet 7 }
```

```
   ptpDelayMechanism OBJECT-TYPE
      SYNTAX  INTEGER {
            e2e(1),           -- The port is configured to use the
   delay request-response mechanism.
            p2p(2),           -- The port is configured to use the peer
   delay mechanism.
            disabled(254)  -- The port does not implement the delay
   mechanism.
            }
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of portDS.delayMechanism shall indicate the
         propagation delay measuring option used by the port
         in computing meanPathDelay.
         The value shall be taken from the enumeration in
         Table 9 of IEEE1588-2008.."
      REFERENCE
         "Clause 8.2.5.4.4, IEEE1588-2008"
      ::= { ptpPortDataSet 8 }


   ptpLogMinPdelayReqInterval OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of portDS.logMinPdelayReqInterval shall be the
         logarithm to the base 2 of the minPdelayReqInterval;
         see 7.7.2.5 of IEEE1588-2008."
      REFERENCE
         "Clause 8.2.5.4.5, IEEE1588-2008"
      ::= { ptpPortDataSet 9 }


   ptpVersionNumber OBJECT-TYPE
      SYNTAX  INTEGER (1..2)
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The value of portDS.versionNumber shall indicate the
         PTP version in use on the port."
      REFERENCE
         "Clause 8.2.5.4.6, IEEE1588-2008"
      DEFVAL { 2 }
      ::= { ptpPortDataSet 10 }


   --
```

```
   --Slave Status data set, containing information on the
   --current status of the clock servo
   --
   ptpSlaveStatusDataSet OBJECT IDENTIFIER   ::= { ptpObjects 7 }

   ptpClockState OBJECT-TYPE
      SYNTAX  INTEGER {
            FreeRun(1),
            Holdover(2),
            Acquisition(3),
            Locked(4) }
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "State of the clock servo loop."
      DEFVAL { FreeRun }
      ::= { ptpSlaveStatusDataSet 1 }


   ptpClockStateDuration OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "Minutes"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Time in current state"
      ::= { ptpSlaveStatusDataSet 2 }


   ptpPhaseCorrection OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "Parts per 1E11"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Current rate correction applied to the output clock
         to provide traceable phase output."
      ::= { ptpSlaveStatusDataSet 3 }


   ptpFrequencyCorrection OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "Parts per 1E11"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
```

```
         "Current rate correction applied to the output clock
         to provide traceable frequency output."
      ::= { ptpSlaveStatusDataSet 4 }


   --
   --Slave Performance data set, containing information on the
   --current estimated performance of the clock
   --
   ptpSlavePerfDataSet OBJECT IDENTIFIER  ::= { ptpObjects 8 }

   ptpResidualPhaseError OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "0.01ns"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Current estimate of the residual phase error."
      ::= { ptpSlavePerfDataSet 1 }


   ptpResidualFreqError OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "Parts per 1E11"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Current estimate of the residual frequency error."
      ::= { ptpSlavePerfDataSet 2 }


   ptpOutputTDEVEstimate OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "0.01ns"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Operational estimate of the clock output's
         peak TDEV stability."
      ::= { ptpSlavePerfDataSet 3 }


   ptpMinimalRTD OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "ns"
```

```
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Minimal round-trip-delay value,
         estimated over a cluster window of 60s"
      ::= { ptpSlavePerfDataSet 4 }


   ptpWeight OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Percentage contribution of the forward flow
         direction used to drive the output.
         Contribution of the reverse direction is the
         inverse of this value."
      ::= { ptpSlavePerfDataSet 5 }


   ptpTrans900Forward OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Number of transient-free seconds in the
         forward direction over a rolling 900s window.
         Transients include loss of sync flow, phase steps
         and pops against dynamic and static thresholds."
      ::= { ptpSlavePerfDataSet 6 }


   ptpTrans900Reverse OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Number of transient-free seconds in the
         reverse direction over a rolling 900s window.
         Transients include loss of sync flow, phase steps
         and pops against dynamic and static thresholds."
      ::= { ptpSlavePerfDataSet 7 }


   ptpTrans3600Forward OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
```

```
         "Number of transient-free seconds in the
         forward direction over a rolling 3600s window.
         Transients include loss of sync flow, phase steps
         and pops against dynamic and static thresholds."
      ::= { ptpSlavePerfDataSet 8 }


   ptpTrans3600Reverse OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Number of transient-free seconds in the
         reverse direction over a rolling 3600s window.
         Transients include loss of sync flow, phase steps
         and pops against dynamic and static thresholds."
      ::= { ptpSlavePerfDataSet 9 }


   ptpOperationalMinTDEVForward OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "0.1ns"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Operational stability estimate in the forward direction
         using the MinTDEV metric."
      ::= { ptpSlavePerfDataSet 10 }


   ptpOperationalMinTDEVReverse OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "0.1ns"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Operational stability estimate in the reverse direction
         using the MinTDEV metric."
      ::= { ptpSlavePerfDataSet 11 }


   ptpOperationalMAFEForward OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Operational accuracy estimate in the forward direction
```

```
         using the MAFE metric."
      ::= { ptpSlavePerfDataSet 12 }


   ptpOperationalMAFEReverse OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Operational accuracy estimate in the reverse direction
         using the MAFE metric."
      ::= { ptpSlavePerfDataSet 13 }


   ptpOperationalMinTimeDisp OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "0.01ns"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Two-way operational accuracy estimate based on the
         MinTDISP (Minimum Time Dispersion) metric. "
      ::= { ptpSlavePerfDataSet 14 }


   --
   --Alarm Status data set, containing information on the
   --current status of alarms within the clock
   --
   ptpAlarmStatusDataSet OBJECT IDENTIFIER   ::= { ptpObjects 9 }

   ptpAlarmRecord OBJECT IDENTIFIER    ::= { ptpAlarmStatusDataSet 1 }

   ptpAlarmID OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "Enumerated list to be defined."
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Enumeration of the type of alarm event."
      ::= { ptpAlarmRecord 1 }


   ptpAlarmDelay OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "s"
```

```
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Configured alarm delay time"
      ::= { ptpAlarmRecord 2 }


   ptpEventFlag OBJECT-TYPE
      SYNTAX  INTEGER {
            CLEAR(0),      -- Associated alarm condition no longer
   exists
            SET(1),        -- Event occurred, associated condition
   exists until cleared
            TRANSIENT(2)   -- Event occurred, with no lasting condition
            }
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Type of event"
      ::= { ptpAlarmRecord 3 }


   ptpAlarmReportingState OBJECT-TYPE
      SYNTAX  TruthValue
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Enables or disables the passive reporting
         of associated alarm."
      ::= { ptpAlarmRecord 4 }


   ptpAlarmReportingProxy OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Enables of disables the active reporting
         of an alarm condition,
         e.g. by generation of an autonomous alarm
         message to the synchronization manager."
      ::= { ptpAlarmRecord 5 }


   ptpAlarmSeverity OBJECT-TYPE
      SYNTAX  INTEGER {
            EVENT(1),
            MINOR(2),
            MAJOR(3),
```

```
            CRITICAL(4) }
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "User-configured alarm severity status."
      ::= { ptpAlarmRecord 6 }


   ptpAlarmRecord0 OBJECT IDENTIFIER   ::= { ptpAlarmStatusDataSet 2 }

   ptpAlarmID0 OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "Enumerated list to be defined."
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Enumeration of the type of alarm event."
      ::= { ptpAlarmRecord0 1 }


   ptpAlarmDelay0 OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "s"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Configured alarm delay time"
      ::= { ptpAlarmRecord0 2 }


   ptpEventFlag0 OBJECT-TYPE
      SYNTAX  INTEGER {
            CLEAR(0),      -- Associated alarm condition no longer
   exists
            SET(1),        -- Event occurred, associated condition
   exists until cleared
            TRANSIENT(2)   -- Event occurred, with no lasting condition
            }
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Type of event"
      ::= { ptpAlarmRecord0 3 }


   ptpAlarmReportingState0 OBJECT-TYPE
      SYNTAX  TruthValue
```

```
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Enables or disables the passive reporting
         of associated alarm."
      ::= { ptpAlarmRecord0 4 }


   ptpAlarmReportingProxy0 OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Enables of disables the active reporting
         of an alarm condition,
         e.g. by generation of an autonomous alarm
         message to the synchronization manager."
      ::= { ptpAlarmRecord0 5 }


   ptpAlarmSeverity0 OBJECT-TYPE
      SYNTAX  INTEGER {
            EVENT(1),
            MINOR(2),
            MAJOR(3),
            CRITICAL(4) }
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "User-configured alarm severity status."
      ::= { ptpAlarmRecord0 6 }


   --
   --Network Statistics data set, containing information on the
   --performance of the network
   --
   ptpNetworkStatisticsDataSet OBJECT IDENTIFIER   ::= { ptpObjects 10
   }

   ptpIPDVObsInterval OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "s"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Observation interval for IPDV statistics."
      ::= { ptpNetworkStatisticsDataSet 1 }
```

```
   ptpIPDVThreshold OBJECT-TYPE
      SYNTAX  INTEGER
      UNITS
         "us"
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Threshold for IPDV measurements."
      ::= { ptpNetworkStatisticsDataSet 2 }


   ptpIPDVSpacingFactor OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Defines the spacing of inter-packet gaps for
         IPDV jitter measurements"
      ::= { ptpNetworkStatisticsDataSet 3 }


   ptpIPDVForwardBelowThreshold OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Percentage of packets in the forward direction
         below the IPDV threshold.
         Measured over the configured observation window."
      ::= { ptpNetworkStatisticsDataSet 4 }


   ptpIPDVReverseBelowThreshold OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Percentage of packets in the reverse direction
         below the IPDV threshold.
         Measured over the configured observation window."
      ::= { ptpNetworkStatisticsDataSet 5 }


   ptpIPDVForwardMaximum OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
```

```
      DESCRIPTION
         "Maximum observed IPDV value in the forward direction.
         Measured over the configured observation window."
      ::= { ptpNetworkStatisticsDataSet 6 }


   ptpIPDVReverseMaximum OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Maximum observed IPDV value in the reverse direction.
         Measured over the configured observation window."
      ::= { ptpNetworkStatisticsDataSet 7 }


   ptpIPDVForwardInterPacketJitter OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Estimate of the inter-packet jitter
         in the forward direction.
         Measured over the configured observation window."
      ::= { ptpNetworkStatisticsDataSet 8 }


   ptpIPDVReverseInterPacketJitter OBJECT-TYPE
      SYNTAX  INTEGER
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "Estimate of the inter-packet jitter
         in the reverse direction.
         Measured over the configured observation window."
      ::= { ptpNetworkStatisticsDataSet 9 }


   -- Notification Types
   --

   ptpEvents OBJECT IDENTIFIER   ::= { ptpRegMIB 2 }

   -- Conformance
   --

   ptpConf OBJECT IDENTIFIER  ::= { ptpRegMIB 3 }

   -- Groups
```

```
   --

   ptpGroups OBJECT IDENTIFIER   ::= { ptpConf 1 }

   -- Compliance
   --

   ptpCompls OBJECT IDENTIFIER   ::= { ptpConf 2 }

   ptpBasicGroup OBJECT-GROUP
      OBJECTS {
      }
      STATUS  current
      DESCRIPTION
         "Basic objects."
      ::= { ptpGroups 1 }

   ptpBasicEvents NOTIFICATION-GROUP
      NOTIFICATIONS {
      }
      STATUS  current
      DESCRIPTION
         "Basic notifications."
      ::= { ptpGroups 2 }

   END
```

5. Security Considerations

To be added.

6. IANA Considerations

To be added.

7. Conclusions

This draft describes in MIB form the management information and performance data that a PTP slave device compliant with the ITU-T's PTP Telecom Profile [G.8265.1] is expected to maintain.

8. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

9. References

9.1. Normative References

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[IEEE1588-2008] "Standard for A Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE1588-2008.

[G.8265.1] "ITU-T PTP Profile for Frequency distribution without timing support from the network ", Draft Recommendation G.8265.1 (work in progress), TD-PLEN-0255-R1, June 2010.

9.2. Informative References

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[Fro2010] Frost, T., and G. Dowd, "Management Requirements for Packet-Based Timing Distribution", draft-frost-tictoc-management-00, work in progress, July 2010.

Author’s Addresses

Tim Frost,
Symmetricom Inc.,
2300 Orchard Parkway,
San Jose,
CA 95131,
USA.
Email: tfrost@symmetricom.com

Greg Dowd,
Symmetricom Inc.,
2300 Orchard Parkway,
San Jose,
CA 95131,
USA.
Email: gdowd@symmetricom.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2011

L. Jin
ZTE
F. Jounay
France Telecom
I. Wijnands
Cisco Systems
N. Leymann
Deutsche Telekom AG

October 22, 2010

Multicast LDP extension for hub & spoke multipoint LSP
draft-jin-jounay-mpls-mldp-hsmp-01.txt

Abstract

This draft introduces a hub & spoke multipoint LSP (short for HSMP LSP), which allows traffic both from root to leaf through P2MP LSP and also leaf to root along the co-routed reverse path. That means traffic entering the HSMP LSP from application/customer at the root node travels downstream, exactly as if it was traveling downstream along a P2MP LSP to each leaf node, and traffic entering the HSMP LSP at any leaf node travels upstream along the tree to the root. A packet traveling upstream should be thought of as being unicast to the root, except that it follows the path of the tree rather than ordinary unicast path.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 25, 2011.

Copyright Notice

Table of Contents

1. Introduction

The point-to-multipoint LSP defined in [I-D. draft-ietf-mpls-ldp-p2mp] allows traffic to transmit from root to several leaf nodes, and multipoint-to-multipoint LSP allows traffic from every node to transmit to every other node. This draft introduces a hub & spoke multipoint LSP (short for HSMP LSP), which allows traffic both from root to leaf through P2MP LSP and also leaf to root along the co-routed reverse path. That means traffic entering the HSMP LSP at the root node travels downstream, exactly as if it was traveling downstream along a P2MP LSP, and traffic entering the HSMP LSP at any other node travels upstream along the tree to the root. A packet traveling upstream should be thought of as being unicast to the root, except that it follows the path of the tree rather than ordinary unicast path.

2. Applications

There are applications that require such kind of LDP based HSMP LSP. According to time synchronization described in [IEEE1588v2], the sync packet and delay request should follow the same path, so as to provide same transmission delay for the two kinds of packets. By using point-to-multipoint technology to transmit these packets will greatly improve the bandwidth usage for above applications. Unfortunately current point-to-multipoint LSP only provides unidirectional path from source to leaf, which cannot fulfill the above new requirement. The main motivation of this draft is to solve the new problem. LDP based HSMP LSP described in this draft provides co-routed reverse path from leaf to root based on current unidirectional point-to-multipoint LSP.

There are two main specific scenarios for timing synchronization based on [IEEE1588v2]: 1. HSMP for phase/time delivery with TCKs. 2. HSMP for phase/time delivery with BCKs. The benefit of using mLDP based HSMP LSP here is to provision dynamically the topology.

Time synchronization is required for accurate quantification of one-way delay as described in [I-D. draft-ietf-mpls-tp-loss-delay]. HSMP LSP can be used to do time synchronization based on [IEEE1588v2] for P2MP LSP or P2MP PW.

The mLDP based HSMP LSP can also be applied in a typical IPTV scenario. There is usually only one location with senders but there are many receiver locations. If IGMP is used for signaling between senders and receivers, the messages from the receivers are travelling only from the leaves to the root (and from root towards leaves) but not from leaf to leaf. In addition traffic from the root is only

replicated towards the leaves. Then leaf node receiving IGMP message (for SSM case) will join HSMP LSP, and send IGMP message upstream to root along HSMP LSP.

Point to multipoint PW described in [I-D. draft-ietf-pwe3-p2mp-pw] requires to setup reverse path from leaf node (referred as egress PE) to root node (referred as ingress PE), if HSMP LSP is used to multiplex P2MP PW, the reverse path can also be multiplexed to HSMP upstream path to avoid setup independent reverse path. In that case, the operational cost will be reduced for maintaining only one HSMP LSP, instead of P2MP LSP and n (number of leaf nodes) P2P reverse LSPs.

3. Terminology

mLDP: Multicast LDP.

P2MP LSP: An LSP that has one Ingress LSR and one or more Egress LSRs.

MP2MP LSP: An LSP that connects a set of nodes, such that traffic sent by any node in the LSP is delivered to all others.

HSMP LSP: hub & spoke multipoint LSP. An LSP allows traffic both from root to leaf through P2MP LSP and also leaf to root along the co-routed reverse path.

4. Setting up HSMP LSP with LDP

HSMP LSP is similar with MP2MP LSP described in [I-D. draft-ietf-mpls-ldp-p2mp], with the difference that the leaf LSRs can only send traffic to root node along the same path of traffic from root node to leaf node.

HSMP LSP consists of a downstream path and upstream path. The downstream path is same as MP2MP LSP, while the upstream path is only from leaf to root node, without communication between leaf and leaf nodes. The transmission of packets from the root node of a HSMP LSP to the receivers is identical to that of a P2MP LSP. Traffic from a leaf node follows the upstream path toward the root node, along the identical path of downstream path.

For setting up the upstream path of a HSMP LSP, ordered mode MUST be used which is same as MP2MP. Ordered mode can guarantee a leaf to start sending packets to root immediately after the upstream path is installed, without being dropped due to an incomplete LSP.

Due to much of same behavior between HSMP LSP and MP2MP LSP, the following sections only describe the difference between the two entities.

4.1. Support for HSMP LSP setup with LDP

HSMP LSP also needs the LDP capabilities [RFC5561] to indicate the supporting for the setup of HSMP LSPs. An implementation supporting the HSMP LSP procedures specified in this document MUST implement the procedures for Capability Parameters in Initialization Messages. Advertisement of the HSMP LSP Capability indicates support of the procedures for HSMP LSP setup.

A new Capability Parameter TLV is defined, the HSMP LSP Capability. Following is the format of the HSMP LSP Capability Parameter.

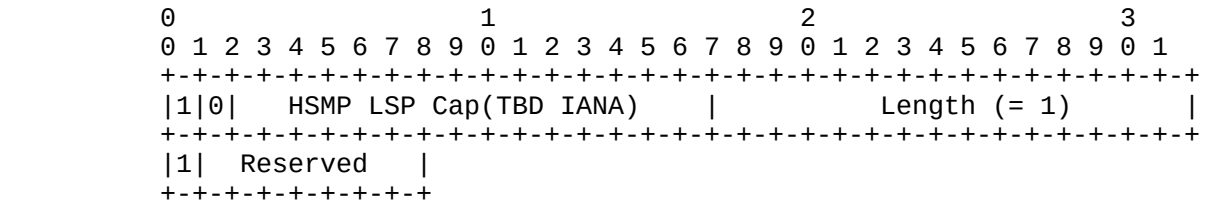

Figure 1

The HSMP LSP capability type is to be assigned by IANA.

4.2. HSMP FEC Elements

Similar as MP2MP LSP, we define two new protocol entities, the HSMP downstream FEC and upstream FEC Element. Both elements will be used as FEC Elements in the FEC TLV. The structure, encoding and error handling for the HSMP downstream and upstream FEC Elements are the same as for the MP2MP FEC Element described in [I-D. draft-ietf-mpls-ldp-p2mp] Section 4.2. The difference is that two additional new FEC types are used: HSMP downstream type (TBD, IANA) and HSMP upstream type (TBD, IANA).

4.3. Using the HSMP FEC Elements

In order to describe the message processing clearly, following defines the processing of the HSMP FEC Elements, which is inherited from [I-D. draft-ietf-mpls-ldp-p2mp] section 4.3.

1. HSMP downstream LSP <X, Y> (or simply downstream <X, Y>): a HSMP

LSP downstream path with root node address X and opaque value Y.

2. HSMP upstream LSP <X, Y> (or simply upstream <X, Y>): a HSMP LSP upstream path for root node address X and opaque value Y which will be used by any of downstream node to send traffic upstream to root node.

3. HSMP downstream FEC Element <X, Y>: a FEC Element with root node address X and opaque value Y used for a downstream HSMP LSP.

4. HSMP upstream FEC Element <X, Y>: a FEC Element with root node address X and opaque value Y used for an upstream HSMP LSP.

5. HSMP-D Label Map <X, Y, L>: A Label Map message with a single HSMP downstream FEC Element <X, Y> and label TLV with label L. Label L MUST be allocated from the per-platform label space of the LSR sending the Label Map Message.

6. HSMP-U Label Map <X, Y, Lu>: A Label Map message with a single HSMP upstream FEC Element <X, Y> and label TLV with label Lu. Label Lu MUST be allocated from the per-platform label space of the LSR sending the Label Map Message.

4.3.1. HSMP LSP Label Map

This section specifies the procedures for originating HSMP Label Map messages and processing received HSMP label map messages for a particular HSMP LSP. The procedure of downstream HSMP LSP is same as that of downstream MP2MP LSP described in [I-D. draft-ietf-mpls-ldp-p2mp]. Under the operation of ordered mode, the upstream LSP will be setup by sending HSMP LSP mapping message with label which is allocated by upstream LSR to its downstream LSR one by one from root to leaf node, installing the upstream forwarding table by every node along the LSP. Detail procedure of upstream HSMP LSP is different with that of upstream MP2MP LSP, and is specified in below section.

All labels discussed here are downstream-assigned [RFC5332] except those which are assigned using the procedures described in section 5.

Determining the upstream LSR for a HSMP LSP <X, Y> follows the procedure for a MP2MP LSP described in [I-D. draft-ietf-mpls-ldp-p2mp] Section 4.3.1.1.

Determining one’s downstream HSMP LSR procedure is much same as defined in [I-D. draft-ietf-mpls-ldp-p2mp] section 4.3.1.2. A LDP peer U which receives a HSMP-D Label Map from a LDP peer D will treat D as downstream HSMP LSR.

Determining the forwarding interface to an LSR has same procedure as defined in [I-D. draft-ietf-mpls-ldp-p2mp] section 2.4.1.2.

4.3.1.1. HSMP LSP leaf node operation

The leaf node operation is same as the operation of MP2MP LSP defined in [I-D. draft-ietf-mpls-ldp-p2mp] section 4.3.1.4, only with different FEC element processing and specified below.

A leaf node Z will send a HSMP-D Label Map <X, Y, L> to U, instead of MP2MP-D Label Map <X, Y, L>. and expects a HSMP-U Label Map <X, Y, Lu> from node U and checks whether it already has forwarding state for upstream <X, Y>. The created forwarding state on leaf node Z is same as the leaf node of MP2MP LSP. Z will push label Lu onto the traffic that Z wants to forward over the HSMP LSP.

4.3.1.2. HSMP LSP transit node operation

Suppose node Z receives a HSMP-D Label Map <X, Y, L> from LSR D, the procedure is same as processing MP2MP-D Label Mapping message defined in [I-D. draft-ietf-mpls-ldp-p2mp] section 4.3.1.5, and the processing protocol entity is HSMP-D label mapping message. The different procedure is specified below.

Node Z checks if upstream LSR U already assigned a label Lu to upstream <X, Y>. If not, transit node Z waits until it receives a HSMP-U Label Map <X, Y, Lu> from LSR U. Once the HSMP-U Label Map is received from LSR U, node Z checks whether it already has forwarding state upstream <X, Y> with incoming label Lu' and outgoing label Lu. If it does, Z sends a HSMP-U Label Map <X, Y, Lu'> to downstream node. If it does not, it allocates a label Lu' and creates a new label swap for Lu' with Label Lu over interface Iu. Interface Iu is determined via the procedures in Section 4.3.1. Node Z determines the downstream HSMP LSR as per Section 4.3.1, and sends a HSMP-U Label Map <X, Y, Lu'> to node D.

Since a packet from any downstream node is forwarded only to the upstream node, the same label (representing the upstream path) can be distributed to all downstream nodes. This differs from the procedures for MPMP LSPs [I-D. draft-ietf-mpls-ldp-p2mp], where a distinct label must be distributed to each downstream node. The forwarding state upstream <X, Y> on node Z will be like this {<Lu'>, <Iu Lu>}. Iu means the upstream interface over which Z receives HSMP-U Label Map <X, Y, Lu> from LSR U. Packets from any downstream interface over which Z send HSMP-U Label Map <X, Y, Lu'> with label Lu' will be forwarded to Iu with label Lu' swap to Lu.

4.3.1.3. HSMP LSP root node operation

Suppose root node Z receives a HSMP-D Label Map <X, Y, L> from node D, the procedure is much same as processing MP2MP-D Label Mapping message defined in [I-D. draft-ietf-mpls-ldp-p2mp] section 4.3.1.6, and the processing protocol entity is HSMP-D label mapping message. The different procedure is specified below.

Node Z checks if it has forwarding state for upstream <X, Y>. If not, Z creates a forwarding state for incoming label Lu' that indicates that Z is the LSP egress. E.g., the forwarding state might specify that the label stack is popped and the packet passed to some specific application. Node Z determines the downstream HSMP LSR as per section 4.3.1, and sends a HSMP-U Label Map <X, Y, Lu'> to node D.

Since Z is the root of the tree, Z will not send a HSMP-D Label Map and will not receive a HSMP-U Label Map.

4.3.2. HSMP LSP Label Withdraw

The HSMP Label Withdraw procedure is much same as MP2MP leaf operation defined in [I-D. draft-ietf-mpls-ldp-p2mp] section 4.3.2, and the processing protocol entities are HSMP FECs. The only difference is process of HSMP-U label release message, which is specified below.

When a transit node Z receives a HSMP-U label release message from downstream node D, Z should check if there are any incoming interface in forwarding state upstream <X, Y>. If all downstream nodes are released and there is no incoming interface, Z should delete the forwarding state upstream <X, Y> and send HSMP-U label release message to its upstream node.

4.3.3. HSMP LSP upstream LSR change

The procedure for changing the upstream LSR is the same as defined in [I-D. draft-ietf-mpls-ldp-p2mp] section 4.3.3, except it is applied to HSMP FECs.

5. HSMP LSP on a LAN

The procedure to process P2MP LSP on a LAN has been described in [I-D. draft-ietf-mpls-ldp-p2mp]. When the LSR forwards a packet downstream on one of those LSPs, the packet's top label must be the "upstream LSR label", and the packet's second label is "LSP label".

When establishing the downstream path of a HSMP LSP, as defined in [I-D.ietf-mpls-ldp-upstream], a label request for a LSP label is send to the upstream LSR. The upstream LSR should send label mapping that contains the LSP label for the downstream HSMP FEC and the upstream LSR context label. At the same time, it must also send label mapping for upstream HSMP FEC to downstream node. Packets sent by the upstream router can be forwarded downstream using this forwarding state based on a two label lookup. Packets traveling upstream need to be forwarded in the direction of the root by using the label allocated by upstream LSR.

6. Redundancy considerations

In some scenario, it is necessary to provide two root nodes for redundancy purpose. One way to implement this is to use two independent HSMP LSPs acting as active/standby. At one time, only one HSMP LSP will be active, and the other will be standby. After detecting the failure of active HSMP LSP, the root and leaf nodes will switch the traffic to the new active HSMP LSP which is switched from former standby LSP. The detail of redundancy mechanism will be for future study.

7. Security Considerations

The same security considerations apply as for the MP2MP LSP described in [I-D. draft-ietf-mpls-ldp-p2mp].

8. IANA Considerations

This document requires allocation of two new LDP FEC Element types:

1. the HSMP-upstream FEC type - requested value 0x09

2. the HSMP-downstream FEC type - requested value 0x10

This document requires the assignment of new code points for the Capability Parameter TLVs, corresponding to the advertisement of the HSMP LSP capabilities. The values requested are:

1. HSMP LSP Capability Parameter - requested value 0x050B

9. Acknowledgement

The author would like to thank Eric Rosen, Fei Su for their valuable

comments.

10. References

10.1. Normative references

[I-D. draft-ietf-mpls-ldp-p2mp]
Minei, I., Kompella, K., and I. Wijnands, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", draft-ietf-mpls-ldp-p2mp (work in progress), October 2009.

[I-D.ietf-mpls-ldp-upstream]
Aggarwal, R. and J. Le Roux, "MPLS Upstream Label Assignment for LDP", draft-ietf-mpls-ldp-upstream-08 (work in progress), July 2010.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036 , October 2007.

[RFC5332] Rosen, E. and R. Aggarwal, "MPLS Multicast Encapsulations", RFC5332 , June 2008.

[RFC5561] Thomas, B., Raza, K., and S. Aggarwal, "LDP Capabilities", RFC5561 , July 2009.

10.2. Informative References

[I-D. draft-ietf-mpls-tp-loss-delay]
Frost, D. and S. Bryant, "Signaling Root-Initiated Point-to-Multipoint Pseudowires using LDP", draft-ietf-mpls-tp-loss-delay-00 (work in progress), July 2010.

[I-D. draft-ietf-pwe3-p2mp-pw]
Martini, L., Jounay, F., Vecchio, G., Delord, S., Jin, L., and L. Ciavaglia, "Signaling Root-Initiated Point-to-Multipoint Pseudowires using LDP", draft-ietf-pwe3-p2mp-pw-00 (work in progress), July 2010.

[IEEE1588v2]
"IEEE standard for a precision clock synchronization protocol for networked measurement and control systems", IEEE1588v2 , March 2008.

Authors' Addresses

Lizhong Jin
ZTE Corporation
889, Bibo Road
Shanghai, 201203, China

Email: lizhong.jin@zte.com.cn

Frederic Jounay
France Telecom
2, avenue Pierre-Marzin
22307 Lannion Cedex, FRANCE

Email: frederic.jounay@orange-ftgroup.com

IJsbrand Wijnands
Cisco Systems, Inc
De kleetlaan 6a
Diegem 1831, Belgium

Email: ice@cisco.com

Nicolai Leymann
Deutsche Telekom AG
Winterfeldtstrasse 21
Berlin 10781

Email: N.Leymann@telekom.de

TICTOC
Internet-Draft
Intended status: Informational
Expires: March 23, 2011

Y(J). Stein
RAD Data Communications

September 19, 2010

Transport of Timing Packets over MPLS
draft-stein-tictoc-mpls-00.txt

Abstract

The TICTOC charter specifies that the working group is concerned with highly accurate time and frequency distribution over native IP and MPLS-enabled IP Packet Switched Networks (PSNs). We discuss here issues specific to MPLS PSNs.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on March 23, 2011.

Table of Contents

1. Motivation

The TICTOC charter specifies that the working group is concerned with highly accurate time and frequency distribution over native IP and MPLS-enabled IP Packet Switched Networks (PSNs). To date, discussions have focused on NTPv4 and 1588-2008 timing distribution using UDP/IP encapsulations. The present document discusses transport of timing packets over MPLS PSNs, and is based on material presented and discussed in previous TICTOC meetings.

We first must address the question as to why special treatment is needed at all for transport of timing packets over MPLS. Timing packets in IP format can certainly be transported over MPLS without the LSRs along the path being aware of them. However, there advantages to being able to recognize, and potentially manipulate, timing packets.

For highly accurate timing distribution, timing packets are required to travel through the PSN with minimal, symmetric, and quasistationary delay, as well as minimal and uncorrelated packet delay variation. Thus, prioritization and symmetric routing of timing packets are minimal requirements. For the most demanding applications, timing distribution mechanisms avail themselves of "on-path support", such as Transparent Clock (TC) overwriting of header fields. None of these can be selectively applied to timing packets unless they can be recognized by the LSR.

2. Encapsulation Options

MPLS as a server layer presently permits three types of clients:
1. MPLS (via label stacking)
2. IP (either IPv4 or IPv6)
3. pseudowires (PWs)

although we can not rule out defining a new client for timing distribution. Taking into account the two defined associated channels that carry non-user traffic, namely the VCCV associated channel for PWs [RFC5085] and the GAL G-ACh defined for MPLS-TP [RFC5586] any proposal for carrying timing packets over MPLS will need to put the timing information in one of the following six formats:
1. a new MPLS client type
2. IP
3. an Ethernet PW

4. a new "timing" pseudowire type
5. a new VCCV channel type
6. a new G-ACh channel type.

We will discuss each of these options in turn.

2.1. Using IP

Since the two main timing distribution protocols have UDP/IP encapsulations, arguably the simplest method of transporting timing packets is in this format. There are several methods for intermediate LSRs to recognize the timing packets :

- Deep Packet Inspection (i.e., peeking under the MPLS stack and identifying well-known port numbers)
- using an arbitrary configured or signaled MPLS label
- using a new reserved MPLS label
- using a specific MPLS Traffic Class (ex-EXP).

It should be noted that there are very few available reserved labels, and that traffic class usage is not standardized.

If an arbitrary label is required to be signaled, then an extension to the signaling protocol will be required. Such an extension will identify the FEC as belonging to a timing flow.

2.2. Using an Ethernet PW

The UDP/IP packets of the timing distribution protocols of the previous subsection may be contained in Ethernet frames, that can be transported over an MPLS network in an Ethernet PW. In addition, IEEE 1588-2008 has a native Ethernet (non-IP) encapsulation.

Methods for identifying timing packets inside an Ethernet PW are similar to those of the previous subsection.

2.3. Defining a New MPLS Client or a new PW Type

IEEE 1588-2008 allows for different transport layers to be defined, with annexes to the standard defining UDP/IPv4, UDP/IPv6, Ethernet, and several other transport networks. It is possible to define a new transport mechanism for MPLS, which entails specifying how to encapsulate a PTP PDU in an MPLS packet. This can be done in the context of the PWE3 architecture (this was proposed in draft-ronc-ptp-mpls-00.txt, now expired) or without reference to that architecture. If the PWE3 architecture is used, then PWE3 features, such as the control word and the control protocol, may be used.

Whether the MPLS encapsulation is a PW or not, the timing packet will be recognized by virtue of its bottom-of-stack label. When additional labels are present, the LSR will need to search for the

label with S=1. This technique is technically a violation of the MPLS architecture, but is presently performed for other purposes, e.g., ECMP avoidance [RFC4928].

The particular label(s) used to signify timing packets may be distributed by manual configuration or a signaling protocol. If the latter is employed, a new FEC type will need to be defined. If a PW is used, a new MPLS Pseudowire Type [RFC4446] will be needed for use in the PWE3 control protocol [RFC4447].

2.4. Using VCCV or the G-ACh

Timing is often distributed in order to enable proper functioning of applications that already define PWs between the end points of the timing flow. In this case, the timing information may be placed in the VCCV associated channel of the application's PW. The associated channel is typically identified by having the same PW (bottom-of-stack) label as the application, but a control word with 0001 in its initial nibble. The timing packets may then be identified by a new channel type, or may use the IP channel type and then would be identified by the well-known port numbers. It is recognized that this requires the LSR to perform relatively deep packet inspection.

If there is no application PW between the timing end points, then we may still use the Generic Associated CHannel (G-ACh) defined in [RFC5586] in a similar manner. The G-ACh is identified by the G-ACh Label (GAL), which is a reserved MPLS label of value 13, at its bottom-of-stack. The format after the GAL is the same as that of VCCV, and thus the considerations of the previous paragraph apply.

3. IANA Considerations

This document requires no IANA actions.

4. References

[RFC4446] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, April 2006.

[RFC4447] Martini, L., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, April 2006.

[RFC4928] Swallow, G., Bryant, S., and L. Andersson, "Avoiding Equal Cost Multipath Treatment in MPLS Networks", BCP 128, RFC 4928, June 2007.

[RFC5085] Nadeau, T. and C. Pignataro, "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, December 2007.

[RFC5586] Bocci, M., Vigoureux, M., and S. Bryant, "MPLS Generic Associated Channel", RFC 5586, June 2009.

Author's Address

Yaakov (Jonathan) Stein
RAD Data Communications
24 Raoul Wallenberg St., Bldg C
Tel Aviv 69719
ISRAEL

Email: yaakov_s@rad.com

TICTOC
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2011

Y. Xu
Huawei Technologies

October 16, 2010

IPsec security for packet based synchronization
draft-xu-tictoc-ipsec-security-for-synchronization-00.txt

Abstract

Cellular networks often use Internet standard technologies to handle synchronization. This document analyses the need for security methods for synchronization messages distributed over the Internet. This document also gives a solution on how to mark the synchronization message when IPSec is implemented in end to end frequency synchronization."

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 19, 2011.

Copyright Notice

Table of Contents

1. Introduction

When transfering timing in internet, a shared infrastructure is used, and hence the path is no longer physically deterministic. It leaves open the possibility to disrupt, corrupt or even spoof the timing flow, where a timing signal purports to come from a higher quality clock than it actually does. In the extreme, this may be used to attack the integrity of the network, to disrupt the synchronization flow, or cause authentication failures. On the other hand, it may be possible for unauthorized users to request service from a clock server. This may overload a clock server and compromise its ability to deliver timing to authorized users.

For the cellular backhaul applications, two kinds of synchronization is needed, one is the recovery of an accurate and stable frequency synchronization signal as a reference for the radio signal (e.g. GSM, UMTS FDD, LTE FDD). In addition to frequency synchronization, phase/time synchronization are also needed in Mobile technologies, This is the case for the TDD technologies such as UMTS TDD,LTE TDD.

Frequency synchronization is normally implemented in an end-to-end scenario where none of the intermediate nodes in the network have to recognize and process the synchronization packets. However In phase/ time synchronization, a hop-by-hop scenario will request intermediate nodes to process the synchronization packets If very accurate phase/ time is needed (e.g. sub-microsecond accuracy).

Femtocell is the typical cellular backhaul application that requires time synchronization. A Femtocell is defined as a wireless base station for deployment in residential environments and is typically connected to the mobile core network via a public broadband connection (eg., DSL modem, cable modem). Femtocell improves cellular network coverage and saves cost for operators. Just like a typical macrocell (larger base station), Femtocells (residential base stations) require a certain level of synchronization (frequency or phase/time) on the air interface, predominantly frequency requirements.

The [3GPP.33.320] specification defines some of the high-level network architecture aspects of a Home NodeB (3G UMTS) and a Home eNodeB (4G LTE). In addition, the Femto Forum organization also provides a network reference model very similar to 3GPP. Both architectures have commonalities as illustrated in Figure 1.

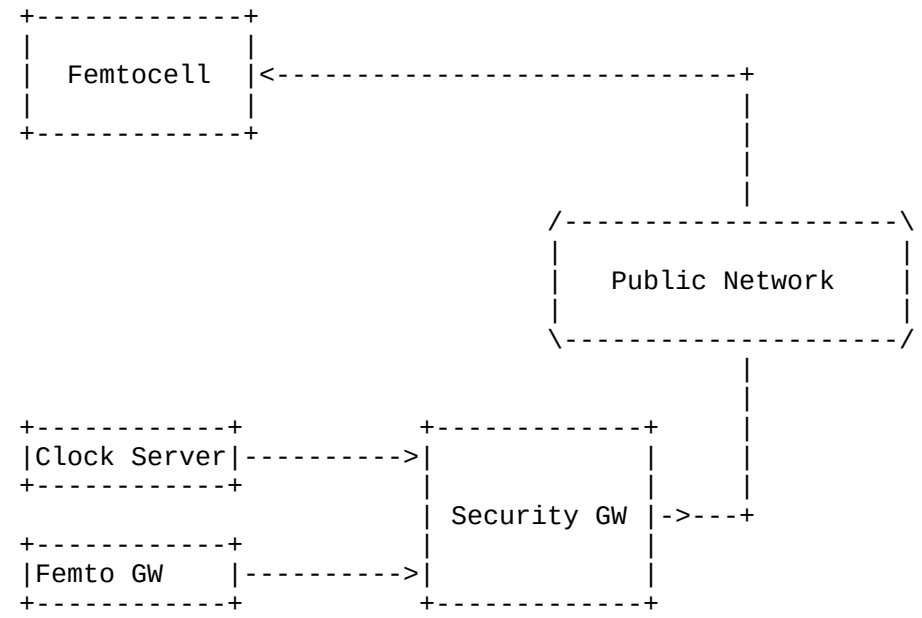

Figure 1. Typical Architecture of a Femtocell Network

The network architecture shows that a public network is used to establish connectivity between Femtocell and core network elements (e.g., Security Gateway, Femto Gateway, Clock server, etc.). With respect to synchronization process, Femtocell will therefore see synchronization messages exchanged over the public network (e.g, Internet). This presents a set of unique challenges for mobile operators.

One challenge involves the security aspects of such the Femto architecture. In both reference models, the communication between Femtocell and Femto Gateway is secured by a mandatory Security Gateway function. The Security Gateway is mandatory since the Femto Gateway and Clock server communicate to Femtocell via a public backhaul broadband connection (also known as the 3GPP iuh interface or Femto Forum Fa interface). The [3GPP.33.320] specification requires that the Femtocell SHALL support receiving time synchronization messages over the secure backhaul link between Femtocell and the Security Gateway, and Femtocell SHALL use IKEv2 protocol to set up at least one IPsec tunnel to protect the traffic with Security Gateway.

This document provides analysis on security requirements for packet-based synchronization and proposes IPsec security solution for end to end frequency synchronization.

2. Terminology used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Security requirements for synchronization

The ITUT [G.8265] specification provides general consideration on synchronization security. Because packet-based timing streams may be observed at different points in the network, there may be cases where timing packets flow across multiple network domains which may introduce specific security requirements. There may also be aspects of security that may be related to both the network (e.g. authentication and/or authorization) and to the synchronization protocol itself. ITUT [G.8265] specification recommends to use existing, standards-based security techniques to help ensure the integrity of the synchronization. Examples may include encryption and/or authentication techniques, or network techniques for separating traffic, such as VLANs or LSPs. Specifically for the performance issue, it may not be possible to implement some security requirements without actually degrading the overall level of timing or system performance. From above analysis, following synchronizations requirements are listed:

1. synchronization client SHOULD be prevented from connecting to rogue clock servers
2. clock servers SHOULD be prevented from providing service to unauthorized synchronization client
3. Security mechanisms to achieve synchronization SHOULD minimize any degradation in performance and this side effect SHOULD be controlled to meet specific synchronization requirements(e.g., Femtocell synchronization)

4. Security mechanism for synchronization

There are mainly two kinds of security mechanism used in current synchronization: authentication-based and encryption-based.

For the authentication-based security mechanism, a shared secret key between the synchronization client and the clock servers is used to compute an authentication code (known as an "Integrity Check Value",

ICV) over the entire message datagram. [IEEE1588] contains an experimental security annex defining an authentication-based approach. This approach also implements a challenge-response mechanism to confirm the creation of any security association (SA) between a clock servers and a synchronization client. A limitation of the process is that no method of sharing the key is proposed in [IEEE1588]. This MUST be handled by other means.

For the encryption-based security mechanism, a shared-key approach is also used. Instead of creating an ICV, the shared key is used to encrypt the contents of the packet completely. The encryption might be performed in the synchronization device itself, or it might be performed in a separate device, e.g. a secure gateway. An example might be where the timing packets have to pass through an encrypted tunnel (e.g. an IPSec tunnel). Full encryption might be required for various reasons. The contents of the packet may be considered secret, such as might be the case where accuracy of the time distribution is being sold as a service. Alternatively, it may be because other traffic from a device is considered secret, and hence it is easier to encrypt all traffic.

IPsec,as a popular security mechanism, is being considered in some mobile applications, especially in case of unsecure backhaul links (e.g. Femtocells, [3GPP.33.320]) being involved. IPsec can provide data source authentication, confidentiality, integrity that is suitable to end to end synchronization without intermediate nodes . For example, if only frequency synchronization is needed, an end-to-end scenario where none of the intermediate nodes in the network have to recognise and process the synchronization packets might be suitable to use IPsec security mechansim. In this case, the synchronization packets will be encrypted if the packed is transported in the IPSec tunnel.

IPsec can meet synchronization rquirement 1 and 2 in section 3, however IPsec still need some enhancement to meet requirement 3 . Normally, device will decrypt IPSec message in IP layer, but in order to improve the synchronization accuracy, some synchronization protocol (e.g. [IEEE1588]) requests to process the synchronization message in hardware, therefore the synchronization device may need to identify synchronization messages in physical layer before the message is decrypted. How to identify the synchronization messages in IPsec becomes the most important issue to keep the synchronization accuracy in IPsec synchronization scenario.

5. ESP format enhancement

As discussed above section, it has advantage to identify whether the

tunnel packets received by synchronization client are the special timing packets or not. This section proposes a solution to identify the timing packets When using IPsec to protect the whole time synchronization message. The main thought is to use time packet identifier which is included in a new defined flexible ESP format to identify whether the received data packet is a timing packet or not.

5.1. Existing ESP format

ESP provides confidentiality, data integrity, access control, and data source authentication to IP datagrams as specified in [RFC4303]. The ESP contains several parts (Figure 2): Security Parameters Index(SPI) and Sequence Number(SQN),ESP Payload,ESP trailer and the ICV. SPI and SQN are used to identity a SA and replay protection respectively. ESP trailer is comprised of Padding, Pad length, Next Header. The integrity scope is from SPI to Next Header. The encryption protection is provided for the Payload Data and ESP trailer. For SPI and SQN, only the authentication of data integrity is provided.

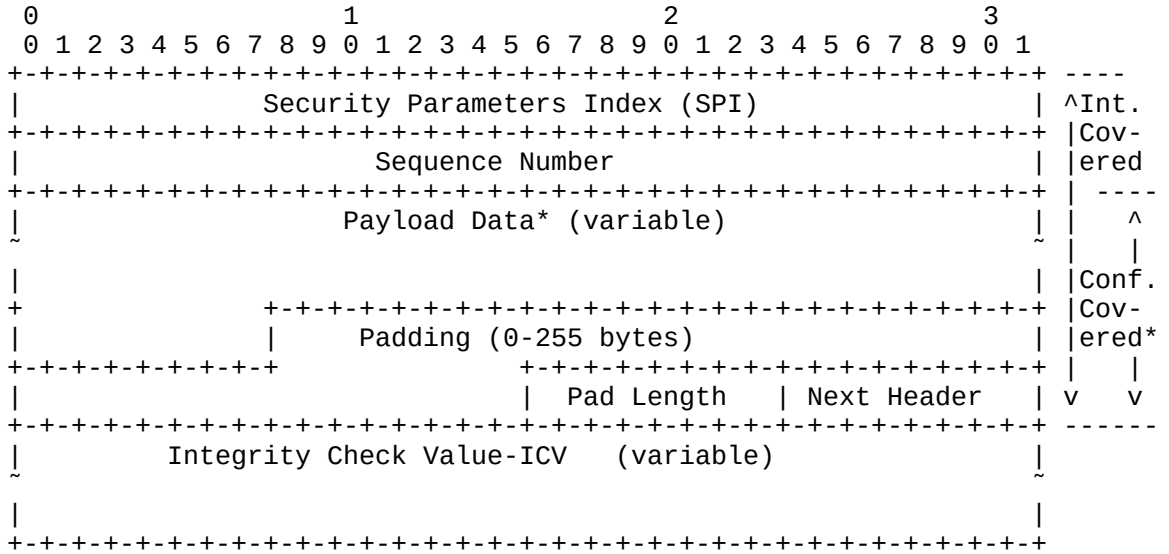

Figure 2. Top-Level Format of an ESP Packet

Except for the fields discussed above, there are no other reserved bits in ESP. However, In the protection of time packets over IPsec scenario, the time packet is encrypted in Payload Data, the receiver could not identify whether it is the time packet or not.

5.2. Flexible ESP format

This documents proposes to define a new flexible ESP format. The new extended ESP format not only contains the fields described in [RFC4303], but also has additional authentication information. The additional authentication information is comprised of ESP special usage flags(one octet zeros),extended data type, extended data length, and Authentication Payload (Figure 3). In the extended ESP additional authentication information, it includes a data type to identify the time packets, and could also identify whether the time packet is the event message or not by addintional time-packet information in Authentication Payload. In addition, the authentication of data integrity for the whole extended Data is provided. The figure of the proposed flexible ESP format is as following:

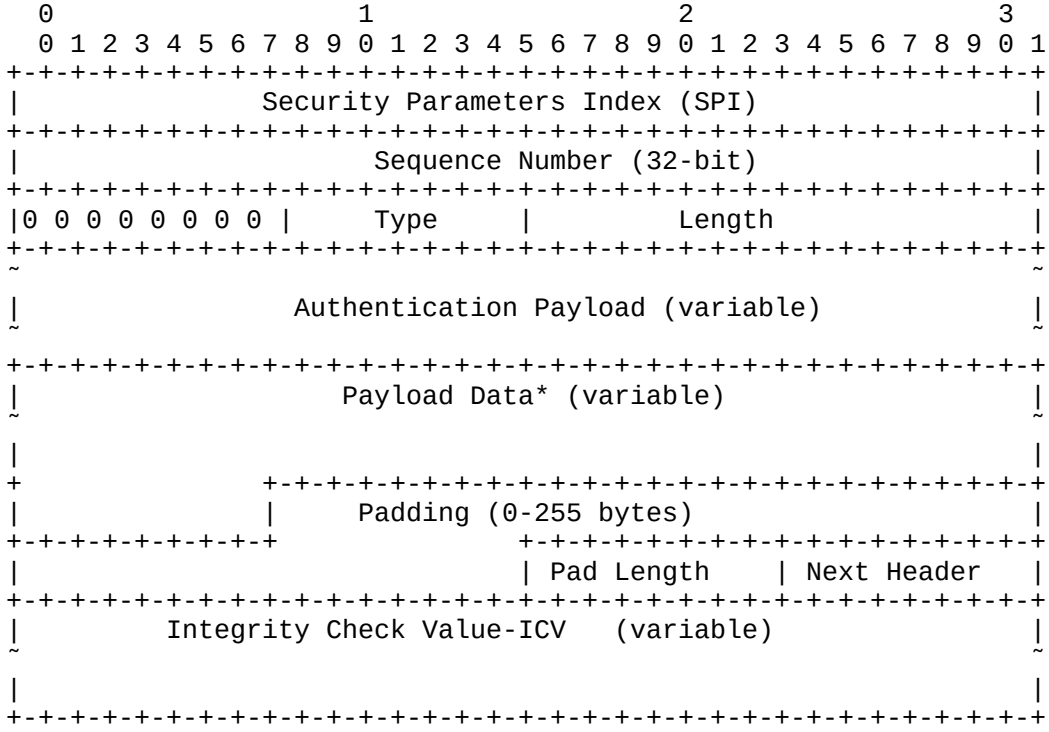

Figure 3. New defined ESP format with 32-bit Sequence Number

o Security Parameters Index(32-bit)-Defined in [RFC4303].

- Sequence Number (32-bit or the extended 64-bit)-Defined in [RFC4303].
- One octet zero bits - The inspection bits, used to distinguish from the existing ESP.
- ED-Type(Extended Data Type (8-bit))- The message type flag in extended additional authentication information which indicates the message type in encryped Payload Data.
- ED-Length(Extended Data Length (16-bit))- The length of extended additional authentication data contains the whole extended additional authentication information.
- Authentication Payload- It contains additional message information, and also contains the information of integrity for extended data, such as, integrity algorithm, and the extended data integrity check value. it is an optional part to provide more information of encrypted messages in Payload Data and also provide authentication of data integrity for extended data, which includes One octet zero bits, Extended Data Type, Extended Data Length and Authentication Payload data.
- Other fields- Defined in [RFC4303].

In Femtocell scenario, as the link between Security Gateway and clock server is normally security path, the message transmitted between them are in plain text. When Security Gateway receives the message, it identifies the time packet at first, then put appropriate value to Data type field to identify the message type in Payload Data, after that, it could put more packet information into Authentication Payload, such as UDP port number or timestamps, then Extended Data Length, Algorithm ID, Extended Data integrity Check value (Figure 4), could also be filled consequently. following figure illustrates on how to use this new flexible ESP format to identify time packet.

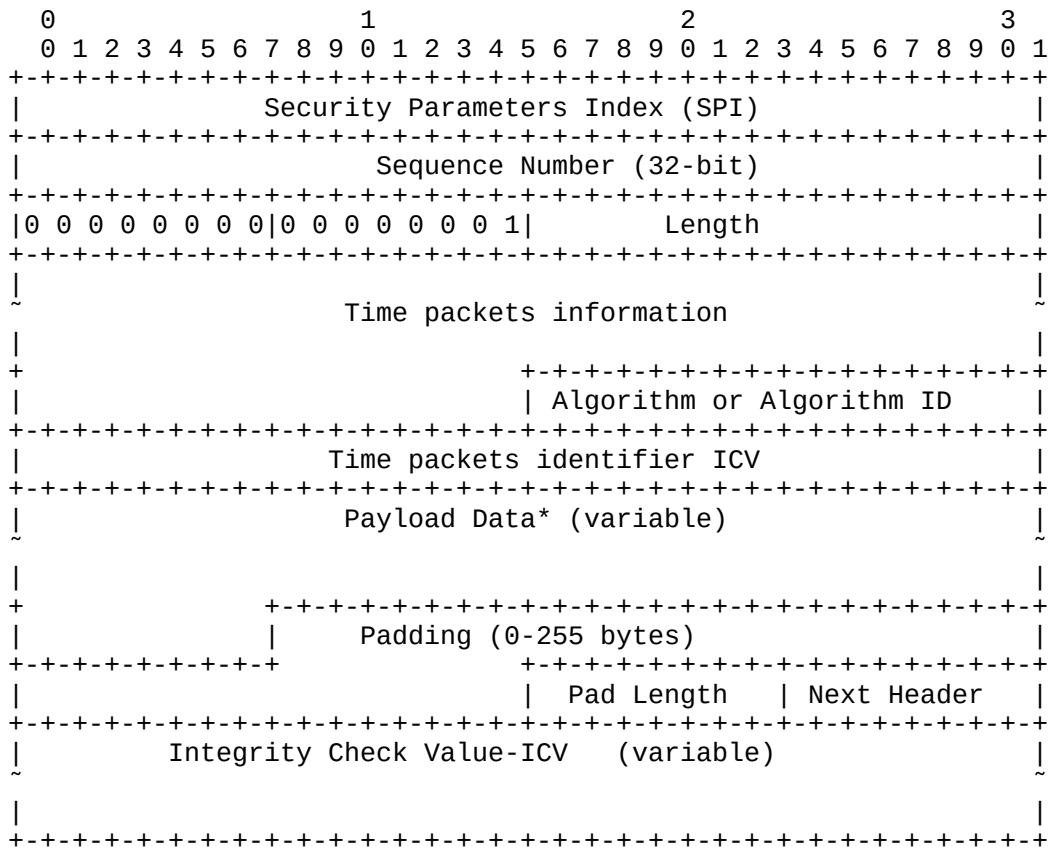

Figure 4. New defined ESP format with 32-bit Sequence Number for time-packet

- o Extended Data Type (8-bit) - The value 0x1 here indicates that the extended context is time packet.
- o Extended Data Length (16-bit)- The length of whole extended additional authentication data
- o Time packets information(variable)- the addintional message information, such as UDP port number or timestamps. It is a part of Authentication payload.
- o Algorithm or Algorithm ID- It indicates which algorithm could be used to generate the extended data ICV. It is a part of Authentication payload.The integrity algorithm negotiated during IKEv2 could be used, also Algorithm ID field in the extended additional authentication data could be marked to indicate the integrity algorithm, such as HMAC- SHA1, HMAC-256, or others. It

is a part of Authentication payload.

- o Extended Data integrity Check value (variable) - Time packets identifier integrity Check value.It is a part of Authentication payload.

Time packets information, Algorithm or Algorithm ID and Extended Data integrity Check value form the Authentication Payload, it is an optinal field and used to guarantee the integrity of transmission. As the integrity protection is only for the Extended Data but not for the whole ESP packet, the time delay of calculation can be decreased. In addition, if the integrity protection is not necessary, this part of security validation could be ignored.

6. Example

In this section, the procedure to identify time packet in Security Gateway scenario is depicted.

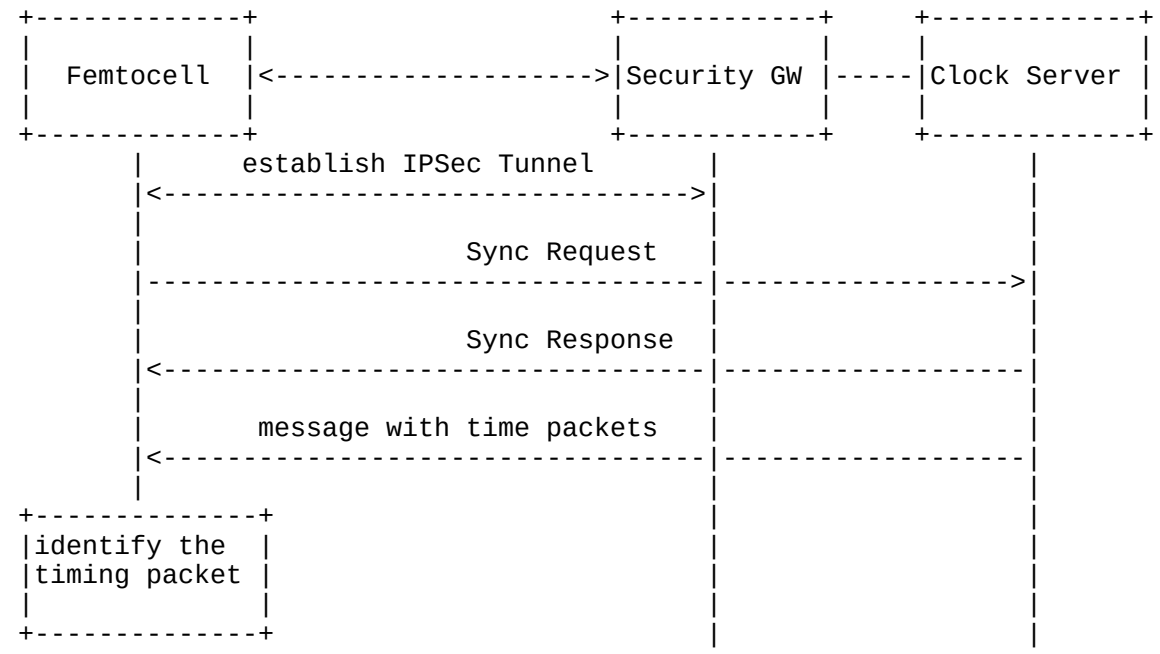

Figure 5. example procedure

In the Security Gateway scenario, The IPsec with tunnel mode is established between Femtocell and Security Gateway. After Femtocell

and Clock server exchange the Sync Request and Sync Response, the clock server will send the time packets to Femtocell to implement frequency synchronization with the protection of IPsec tunnel. When Femtocell receives the message, it can identify whether it is time packet, and can also identify whether the time packet is the event message by the time packet information in the unencryped field as defined in the new ESP format. If the message is time packet and identifies that it is the event message, Femtocell will do special process for the event message, such as recording the message receiving time. On the server side, When Security Gateway receives the message, it identifies the time packet at first, then put appropriate value to Data type field to identify the message type in Payload Data, after that, it could put more packet information into Authentication Payload, such as UDP port number or timestamps, then Extended Data Length, Algorithm ID, Extended Data integrity Check value, could also be filled consequently.

7. IPv4/v6 consideration for IPsec based sychronization

IPsec is a security mechanism used both for IPv4 and IPv6, and ESP-based solution has no impact on the IPv4 header and makes the transition/migration from IPv4 to IPv6 seamless.

8. Security Considerations

This protocol variation inherits all the security properties of regular ESP as described in [RFC4303].

This document defines a new flexible ESP format, which contains Extended Data Type Extended Data Length and additional authentication paylaod. The authentication of data integrity for the extended data is provided, and the data type is carried in the unencryption part. Therefore the receiver could identify whether the receiving messages are time packets or not.

9. IANA Considerations

There have been no IANA considerations so far in this document.

10. Acknowledgments

The authors appreciate the valuable work and contribution done to this document by Marcus Wong.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

11.2. Informative References

[3GPP.33.320] 3GPP, "Security of Home Node B (HNB) / Home evolved Node B (HeNB)", 3GPP TS 33.320 10.0.0, October 2010.

[G.8265] IEEE, "Architecture and requirements for packet based frequency delivery", V0.2 June 2010.

[IEEE1588] IEEE, "Standard for A Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588-2008.

Author’s Address

Yixian Xu
Huawei Technologies
Huawei Building, Xinxi Road No.3
Haidian District, Beijing 100085
P. R. China

Phone: +86-10-82836300
Email: xuyixian@huawei.com